Proceedings of the EXPERT Scientific and Technological Workshop

Hernani Costa, Anna Zaretskaya, Gloria Corpas Pastor,
Lucia Specia, Miriam Seghiri (eds.)

Malaga, 26^{th} and 27^{th} June 2015

ISBN 978-2-9700736-6-6

Scientific Committee

Alessandro Cattelan (Translated, Italy)

Carla Parra (Hermes, Spain)

Constantin Orasan (University of Wolverhampton, UK)

Eduard Barbu (Translated, Italy)

Gideon Maillette de Buy Wenniger (University of Amsterdam, Netherlands)

John Judge (Dublin City University, Ireland)

Jorge Leiva (University of Malaga, Spain)

Josef van Genabith (University of Saarland, Germany)

Kashif Shah (University of Sheffield, UK)

Khalil Sima'an (University of Amsterdam, Netherlands)

Lucia Specia (University of Sheffield, UK)

Manuel Arcedillo (Hermes, Spain)

Manuel Herranz (Pangeanic, Spain)

Marcos Zampieri (University of Saarland, Germany)

Qun Liu (Dublin City University, Ireland)

Ruslan Mitkov (University of Wolverhampton, UK)

Table of Contents

The Use of Machine Translation among Professional Translators

Anna Zaretskaya
University of Malaga, Spain
annazar@uma.es

Abstract

This paper presents results of a user survey for professional translators, which was aimed at identifying their needs regarding translation technologies. It focuses specifically on machine translation (MT), which user groups are more likely to adopt it and how they perceive technological advancements in this field. Based on the data, some connections could be made between the use of machine translation and translators' domain of specialisation. However, future advancements of MT technology are perceived independently of the domain. Translators with advanced knowledge in IT tend to use MT more than the ones with less IT skills. Similarly, education in IT also has an effect on MT usage rate. Finally, we identified that more freelance translators who work with an agency tend to use MT more than those who work without an agency.

1 Introduction

From translators' point of view, translation tools are computer software that aims to facilitate their work, make the project delivery faster and easier, save their time by solving easier tasks in an automatised way and allow them to concentrate on more challenging and creative parts of the translation process, and finally, to increase their income (Bowker and Pastor, 2014). In practice, the amount of tools available also makes translator's life difficult, as they have to decide which tools are useful for them and how to integrate them in their workflow. Thus, machine translation (MT) services available nowadays evoke contradictory attitudes among translators. On one hand they are cheap and easy to use, and therefore can provide a fast draft translation that only needs some editing. On the other hand, the quality of translation is not satisfactory enough for all domains and languages even as a draft, so many translators find them useless for their job and prefer to make translations from scratch. In addition, there is an increasing concern related to the security of the information translated on the Web, and many translators who do like working with MT are imposed to sign confidentiality agreements with their clients for not using any such service.

There is a common opinion that translators perceive the MT technology as a threat for various reasons. Firstly, the use of MT transforms their role from translators into post-editors, thus significantly reducing the creative component of translator's job. Secondly, it also implies lower rates, as post-editing is normally lower remunerated than translation. And finally, it is thought that with the development of high quality machine translation, the translator profession will no longer exist in the same form as we have it now, it will disappear completely or in the best case transform into something similar to project manager.

In this paper we will present the results of a user survey for professional translators conducted with the purpose of identifying their requirements regarding various translation technologies, as well as their current working practices, i.e. which tools and resources they use and how they do it, degree of satisfaction with these technologies concerning the quality of output, learning curve, offered functionalities, productivity and income increase, levels of awareness of different types of technologies available, possible reasons for low usage rate for different tools and missed opportunities for reaching potential users, and their overall attitude towards current technology-related industry trends. The paper

focuses on machine translation and presents the survey findings on the aspects of user profile that are related to the use of MT.

The structure of the paper is as follows. In section 2 we make a review of previous user surveys on translation technologies. Section 3 describes the survey and the population of participants. In Section 4 we discuss general findings on MT usage and attitudes, as well as usage rate with resource-rich and resource-poor languages (Section 4.1), with different domains (Section 4.2), how it is used by translators who have different levels of computer competence (Section 4.3) and education in IT (Section 4.4), compare translators who work with an agency or independently (Section 4.5). Finally, in Section 5 the results are summarised and discussed.

2 Findings on MT from previous surveys in translation research and industry

In the recent years a number of user surveys on technological and user-related aspects of the translation industry have been conducted. Some of them focused specifically on MT. By a way of example, the QTLaunchPad survey (Doherty et al., 2013), carried out in 2013, was specifically focused on the use of MT in the industry. Under 500 translation services buyers and vendors gave their opinion on translation quality methods and technologies. Apart from questions on translation quality assessment, the respondents were asked about their adoption of MT systems. Over one third of the respondents reported that they were currently using MT, while 13% stated that their businesses were currently not using MT, but were planning to do so. However, 28% of the respondents said they did not use MT and have no plans to start doing so. The most popular type of MT systems is statistical machine translation (SMT), which was mentioned by over a half of MT users. Hybrid MT (HMT) was used by 36%, followed by rule-based systems (RBMT) with 22%. One third of all the MT adopters use external online systems like Google Translate, BabelFish and Bing. Users of off-the-shelf MT systems were asked whether they performed any kind of customisation of the MT systems, with 84% positive replies. Popular modifications lie in areas of terminology (61%), in the use of additional domain-specific corpora (32%), and by providing tailor-made linguistic rules (21%), while 16% did not implement any modifications. Regarding the quality of MT output, 69% stated that less than half of their outbound translation requirements were satisfied with MT, while 12% think they can use more than half of MT translated content and 4% use MT for all their content. Opinions of the respondents on the quality of translation performed by the systems were predominantly positive, 43% rated it as fair, 41% as good, and 2% as excellent. Only 7% of answers rated it as poor.

	(Doherty et al., 2013)	**(Torres Domínguez, 2012)**
Number of participants	500	509
Currently using MT	34%	21%
Not using, but planning to	35%	16%
Not using and not planning to	29%	24%
SMT	51%	48%
HMT	36%	34%
RBMT	22%	18%

Table 1: Compared results of two surveys by Doherty et al. (2013) and Torres Domínguez (2012).

The Use of Translation Technologies survey, reported one year before (Torres Domínguez, 2012), collected answers from various participants involved in the translation workflow, such as translators, project managers, reviewers, DTP specialists, linguists, etc. MT applications were used considerably less compared to the QTLaunchPad survey. Consider some of the results of the two surveys compared in Table 1. Only 21% were using MT at the time of the survey, and 16% were planning to use it. About a quarter of translators did not use it, and 7.5% were not familiar with MT at all. This can be caused by the differences in the population of the two surveys, but also can be an indication that the usage rate is increasing. Concerns about the quality of translation performed by MT systems seem to

be the main reason for reluctance to use them. And even translators who used MT mostly evaluated its output quality as flexible (54%), and 26% used MT just to get the gist of the text. Despite the quality concerns, more than half of the MT users believed that it helps save working time and effort. Only 39% thought it accelerates delivery, for 35% it helps maintain terminology consistency, and 32% mentioned cost savings. The most widespread type of MT systems among respondents was SMT systems (used by 48%), 34% mentioned HMT systems, 28% used example-based (EBMT) and 18% RBMT systems. The prevalent service was Google Translate with 55%, Systran was used by 17.5%, BABYLON by 15%, and Moses by 11%.

Despite the low usage rate among translators, the importance of MT technology was realised by many translators already in 2011, according to the survey conducted by Trad Online (2011). It focused on the changes in translation industry caused by arising of new technologies, translators attitudes and expectations regarding these changes, as well as evolution of technology as a whole. Among 1330 respondents the vast majority were freelance translators and interpreters. A big part of the respondents (48%) believed that automated translation will have impact on how translators do business in the near future, while 26% thought there wouldnt be any changes in the next 3-5 years related to MT, and 22% think were foreseeing significant changes coming along.

An earlier survey, which also aimed at shedding light on the use of machine translation, was carried out by SDL (SDL, 2009). The answers were received from 228 participants from translation companies all over the world. The results revealed that 17% respondents (a slightly lower rate compared to the 2012 Translation Technologies survey discussed above) were using MT and 28% used or were planning to use MT. The major concern reported by 76% which prevents respondents from using MT is the quality of output. Due to the quality concerns, 37% of respondents would not use a public Internet-based service while 28% consider using it inappropriate. The type of documents that is most frequently translated with MT is technical texts (60%). In order to improve the quality, 57% of participants are more likely to adopt MT when coupled with human post-editing, while 30% indicated that they were already post-editing or had imminent plans to do so.

Earlier surveys related to the subject of MT usage include the Gilbane Group (2009) survey on multilingual product content which aimed investigating how global product content is handled in multilingual organisations; Lagoudaki's (2006) survey on translation memory (TM) systems, which also covers the topic of MT functionality in TM software; the survey on translation technology and UK freelance translators by Fulford and Granell-Zafra (2005).

To summarise, these surveys provide some important insights on the usage of MT in the industry. On one hand, the MT usage rate seems to be increasing over the years. However, low quality of translation is still the main reason why the majority of translators remain reluctant to this technology. At the same time, many of the MT users admit that it increases their efficiency in terms of fast delivery and time saving. Having in mind this contradiction, it is interesting to find out what user-related factors might be connected with their attitudes towards MT and its usage rate among different groups of translators.

3 Survey design and population

In this paper we suggest some of the potential factors that can be related to the usage of MT based on the data obtained from the survey "Computer Tools for Translators: Users' Needs". The survey was composed of separate sections, where the first section concerned the user profile, the second section included general questions on the use of technologies, and the rest of the sections were focused on specific types of tools, such as machine translation, translation memories, corpora compilation and terminology extraction, and also covered some aspects related to quality assurance tools and various web-based lexicographical resources. For this research we mostly used the data from the machine translation section, while the rest of the data will be used on further stages of our research.

The survey was built online and the link to it was distributed through translation companies, mailing lists and social media groups for translators, translation blogs and translation associations. One of the challenges during this stage was to attract a sufficient number of participants to be able to obtain statistically significant results. Fortunately, we had a chance to access a large database of freelance

translators through one of the partner companies, which helped us receive a very large number of responses: we received 736 completed responses and 1304 responses in total. This indicates a high response rate but a low completion rate, which is mainly due to the large size of the questionnaire. The participants responded actively and many provided feedback and comments.

As to the participants' profile, they originated from 88 different countries, about a half of them being from Italy, Spain, Germany, USA, UK, Brazil, Belgium, Finland and Portugal. The vast majority of translators worked as freelancers. The two largest subgroups were freelancers who had an agency but also worked independently apart, and freelancers who only worked independently. Only 12% just worked with an agency, 3% as in-house translators in a translation company and other 3% in a non-translation company.

The questionnaire data was analysed in three steps. The first step included descriptive analysis of the quantitative data, i.e. the answers were described with the help percentages and graphs in order to show the data distribution. These results are summarised in (Zaretskaya et al., 2015). Secondly, we analysed the qualitative data obtained from the open-ended questions (i.e. respondents' comments in their own words) using a coding methodology described in (Auerbach and Silverstein, 2003). And finally, we performed bivariate analysis to find dependencies between pairs of questions (Lee and Forthofer, 2006).

4 Use of MT

In this section, we present statistics regarding the use of MT and analyse possible factors that might be related to it. In total, MT systems were used by 36% of respondents, while almost the same number (38%) were not using any MT and were not planning to use them in the future. A smaller percentage (15%) did not reject this technology completely claiming that they were planning to use it in the future, and 11% used it before but abandoned it afterwards (Figure 1).

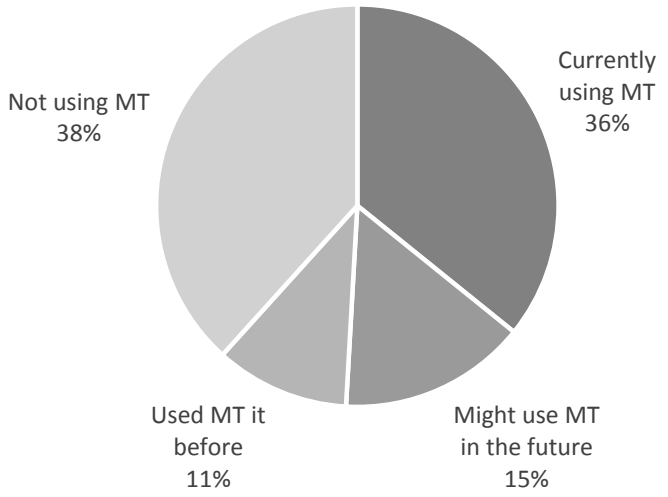


Figure 1: Use of MT.

Most respondents who did not work with MT reported it being due to bad quality (67%) and because they did not find it useful (35%). In addition, some respondents commented that they were not allowed to use MT due to the clients' requirements.

In order to investigate how translators see the future of the MT technology and its influence on their profession, we asked participants if they thought they could benefit from high quality machine translation, i.e. a system that would translate with almost 100% accuracy. A rather positive result was obtained with 74% of the respondents to this question answering "yes".

Furthermore, respondents were asked to provide their comments explaining their opinion in a text box. We received 158 comments which were collected and coded. Coding is a necessary step in the analysis of qualitative data (Auerbach and Silverstein, 2003; Basit, 2003), which consists in annotating

every item of obtained data according to the topic or topics it is related to, and following a certain hierarchy. As a result, 16 classes were identified according to the concepts mentioned in the comments. A common opinion among respondents (expressed by 48 participants) was that high-quality MT will never exist, as "a machine can never be a human". However, the possible benefits that translators can see are time saving (36), efficiency or, in other words, larger volumes delivered in less time (25), and effort saving (14). Additionally, good MT can improve translation quality for 6 respondents, consistency for 5 respondents, increase their income (4) and give an opportunity to concentrate on more challenging "high-value parts that only a human can translate" (2). On the other hand, those opposed to machine translation argue that its development will make them lose their jobs (10) or lead to lower wages (9) and to general devaluation of their profession (3). Two participants were convinced that even when working with a high-quality system "human should be in control".

To summarise, the majority of respondents did not use MT at the time of the survey, even though the usage rate was higher compared to previous surveys. Most translators thought that could benefit from advancements in MT. Many of them believed that perfect automatic translation will never be achieved, therefore there is no threat for their profession, but a good MT system can increase their productivity and income. However, some translators have expressed concerns about their future as professionals related to the development of MT.

4.1 MT and languages

We further investigated more in detail the user profiles of MT-users and non-users and tried to find dependencies which might help us draw conclusions about translators' attitudes towards MT. First, we considered the translators' working languages and whether or not they worked with machine translation. Low quality of the MT output is one of the main reasons why translators disregard this technology. Most free online systems are based on statistical methods, i.e. their quality depends substantially on the amount of data available. Therefore, our initial hypothesis was that there is a dependency between the use of MT and the translator's working languages: if translators work with rare or resource-poor languages, it is less likely for them to use MT because the quality of the output is lower.

In order to test our hypothesis, we first divided all the languages into two groups: resource-rich languages and rare or resource-poor languages. The first group included English, Spanish, German, French, Italian, Chinese, Dutch, Portuguese and Japanese. The second groups included Afrikaans, Albanian, Arabic, Armenian, Bengali, Bulgarian, Croatian, Czech, Danish, Estonian, Finnish, Georgian, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Irish, Javanese, Korean, Latvian, Lithuanian, Macedonian, Malay, Maltese, Norwegian, Persian, Polish, Romanian, Russian, Serbian, Slovak, Slovenian, Swedish, Thai, Turkish, Ukrainian, Uzbek and Vietnamese.

Then, we divided all the survey respondents into four groups according to the languages they worked with. Group one includes translators with both source and target language being resource-rich (R-R); group two with both languages resource-poor (P-P), group three and four work with one resource-rich and one resource-poor language (R-P and P-R correspondingly). Also, to simplify the presentation of the data, we consider only two respondent groups according to their use of MT: users and non-users. The users are translators who were currently using MT or had used it before, and the non-user group are the ones who never used it. We compared the number of MT users and non-users for each language-pair group (Table 2). Even though the number of MT-users is equal to the number of non-users for translators who work with two resource-poor languages (which might be due to small size of this population group), we can see a bigger difference in the number of users and non-users for the group where the target language is resource-poor, compared to the R-R group and the P-R group.

We used the chi-square (χ^2) test for independence to verify whether there is a dependency between the use of MT and the working languages. The chi-square test is commonly used for survey data analysis to verify whether two categorical variables can be related (Rao and Scott, 1981; Lee and Forthofer, 2006). We achieved the (χ^2) value of 1.816 with the degree of freedom df = 3 and the p-value of 0.6115. The high p-value (much bigger than 0.05, which corresponds to the 95% level of confidence) indicates that there is no dependency between the languages and the use of MT, which contradicts our initial intuition.

Language pair	MT users	Non-users
R-R	198	206
R-P	102	132
P-R	18	21
P-P	8	8

Table 2: Use of MT with resource-rich and resource-poor languages.

This can be an indication that translators make their choices independently of their working languages and that, even though the quality of automatic translation is worse with some language pairs, it is still suitable for some users and some working scenarios. And on the contrary, even with popular languages for which the quality of MT should in general be better, it is still not enough for some translators to adopt this technology. Consequently, this allows us to suppose that there are more important factors influencing translators' decision to use MT, such as working habits, knowledge of technologies, type of employment, domain of specialisation, among others. In the following sections we consider some of them.

4.2 MT with different domains of specialisation

It is thought to be common for translators to work with texts that fall within a specific subject or subjects that they are more familiar with, e.g. medical texts, literature, legal documents, etc. However, it turns out that not many translators limit their field of specialisation only by one domain. The respondents of the survey were able to choose more than just one domain of specialisation and it turned out that, indeed, many of them preferred to work in more than only one domain, with the average number of domains chosen by one participant accounting to 5.16. This seems logical, since humans tend to prefer work that involves diversity and variation rather than constant repetition, so many translators find it more rewarding to work with several domains rather than being limited by just one. On the other hand, it may also imposed by the industry requirements, as companies often prefer to work with the same translators who already proved to be professional instead of hiring a new different translator for each domain.

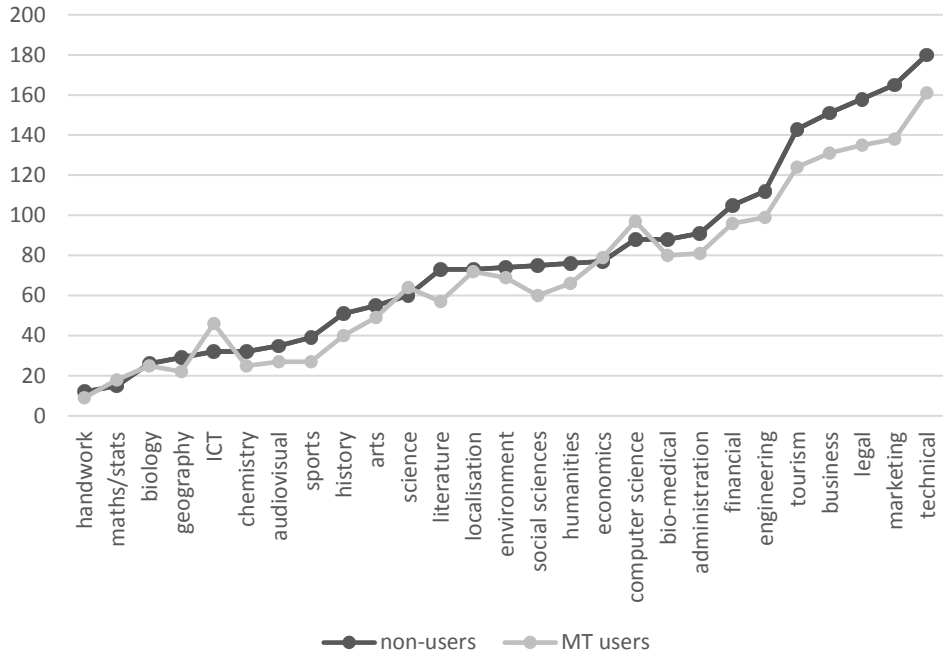


Figure 2: Number of MT-users and non-users with different domains.

A previous survey on translators' needs by Lagoudaki (2008:161) revealed that there is a dependency between the domain of specialisation and the usage rate of MT. Translators who work mostly with

technical and specialised texts with restricted vocabulary and less idiomatic expressions are more likely to resort to automatic translation than translators working with more creative texts, for instance, from the marketing domain.

Our results are in line with this statement, which is illustrated by Figure 2. Again, we considered two groups of users and non-users of MT. As we can see, the number of MT users is generally lower than the number non-users, except for some domains where the users' line approaches or overpasses the non-users' line. These domains were mathematics and statistics, biology, ICT (information and communications technology), science, games and software localisation, and computer science. However, within engineering and technical domains MT is still less used. On the contrary, the difference between the two lines increases for the literature, sports, and the social sciences domains.

Considering the results discussed above, we assumed that translators who see developments in MT as a positive tendency mostly work with technical and specific domains, because high quality machine translation would allow them to avoid translating the same terms and repetitions, compared to translators of more creative or literary texts, who might want to preserve the creative component of their work instead of leaving everything to the automatic translation. To verify this hypothesis, we correlated the two corresponding variables (Figure 3).

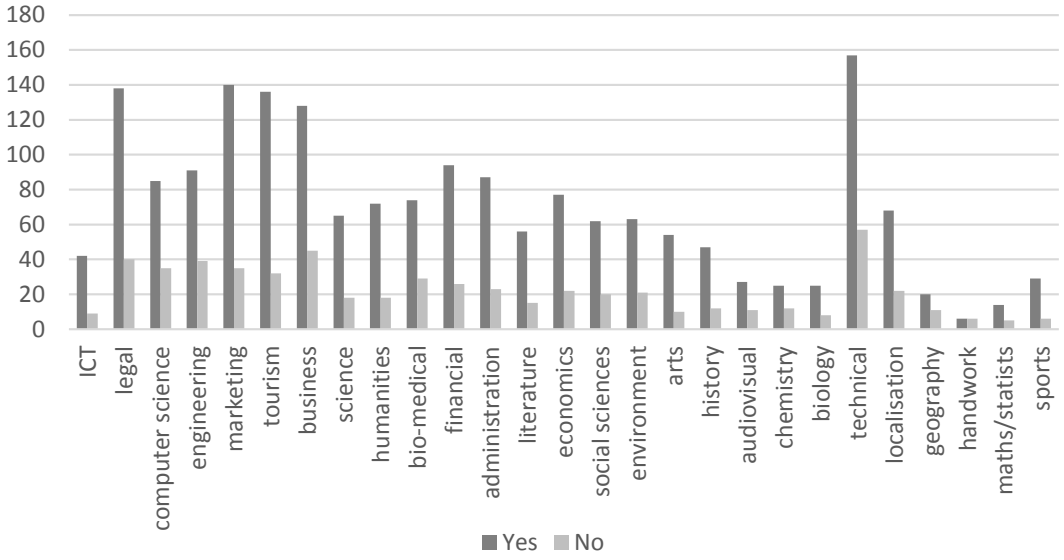


Figure 3: Could you benefit from high quality machine translation?

This turned out to be partially true, but only for some of the domains. For instance, translators working in technical domain are very likely to have a positive attitude towards developments in MT, as well as legal domain. But it is also true for marketing, tourism, and business domains, which contradicts our initial assumption. However, this proves that translators are willing to use MT not only with some specific domains where it already performs considerable well, but with all domains. The fact that the quality of MT translation is much lower for some domains does not necessarily lead to translators' rejecting this technology completely. In addition, these results show that translators in all domains are more likely to be disposed to make their contribution to the development of MT and, for instance, provide their feedback on MT output.

4.3 Use of MT and computer competence

The participants of the survey were asked to evaluate their own computer competence as advanced, experienced, average or poor (Table 3).

We can see that the number of current and previous MT-users (group 1 and 2) decreases with their auto-estimated computer competence. In the other two groups, i.e. the non-users, there are more experienced users rather than advanced, and the numbers of translators with average and poor competence decrease

Use of MT	Advanced	Experienced	Average	Poor
1. Currently using MT	134	99	23	0
2. Used MT before	46	25	5	1
3. Currently no using MT, but might in the future	38	49	16	3
4. Not using and not planning to	112	126	30	3

Table 3: Use of MT and users' computer competence.

again. This can indicate that experience and knowledge in IT allow translators incorporate MT in their workflow in a beneficial way, for instance, through integration with their CAT tool. The correlation with the computer competence was verified by the chi square test, which yielded a low p-value of 1.19e-14 with the degree of freedom df=16 – a low p-value (much below the 0.05 threshold) indicates that there is a dependency between the two variables.

4.4 Education and training in IT

All survey participants were asked whether they had received any education or training in IT and computer science. The majority of all participants (61%) had some training, which shows that translators have great interest in technologies and are motivated to learn how to leverage the variety of available tools. The respondents could choose more than one option among "BSc in IT", "MSc in IT", "PhD in IT", "Specialised courses, seminars, workshops on IT", "Specialised courses on computer-assisted translation (CAT) tools", and "None".

Only a small number of participants appeared to have a university degree in IT and computer science (see Table 4). For instance, from all respondents who had a bachelor's degree in IT, 13 were MT users at the time, 5 did not use MT, 4 were planning to use it in the future, and another 5 had used MT before. On the contrary, among the holders of a master's degree, the highest number did not use MT at all (11). Finally, out of 5 doctors in computer science 4 were MT users.

	Currently using MT	Do not use MT	Might use MT in the future	Used MT before
BSc	13	5	4	5
MSc	8	11	3	4
PhD	4	0	1	0
Courses in IT	117	105	45	42
Courses on CAT	85	61	37	32
None	94	127	40	20

Table 4: Use of MT and education in IT.

Much more translators had done courses in IT and specialised courses on CAT tools, and a significant number did not have any training in IT. Figure 4 shows the distribution of the most popular types of training for each MT user group. We can make two observations based on this chart. Firstly, most of the current users of MT had done courses in IT, while most of the non-user group did not have any training in IT. In addition, the number of translators who had done courses on CAT tools is lower for the group of non-users. Therefore, a conclusion can be made that courses on IT and CAT tools might positively influence the MT usage rate among translators.

We considered similar correlations with the translators' perception of developments in MT. For this, two groups were compared: the group of respondents who thought they could benefit from high quality MT (Figure 5a), and the group who thought they could not (Figure 5b). The two figures seem very similar, even though they differ significantly in size (the groups who favoured high quality MT is much larger). This means that both groups had similar education in IT and it does not have any impact on their attitude towards advancements in machine translation. To conclude, training in IT and computer

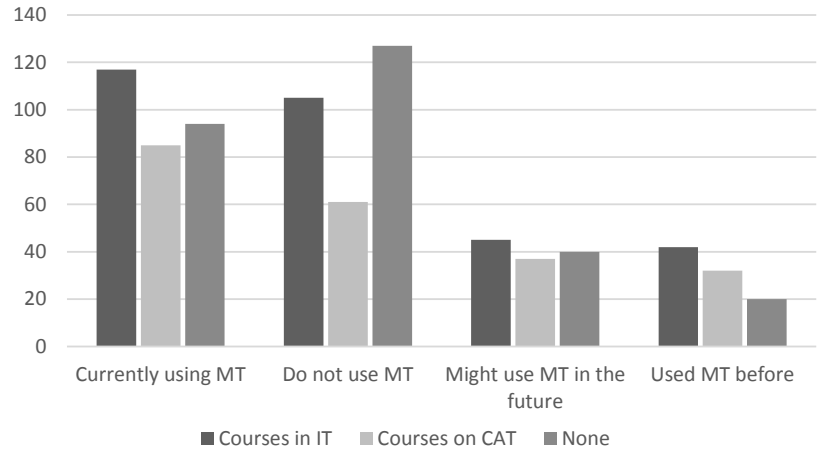


Figure 4: Use of MT by translators who did courses in IT and CAT tools.

science (courses and seminars in particular) can increase MT usage rate among translators, but not their perception of how this technology might affect them in the future.

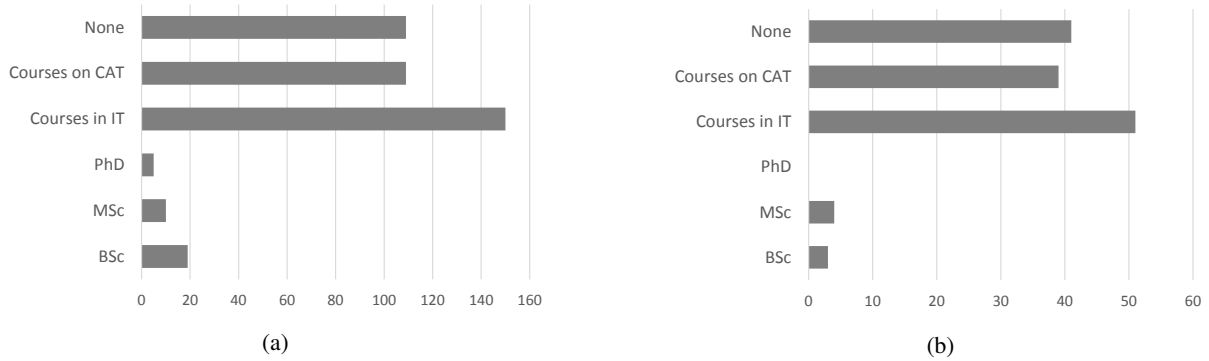


Figure 5: Education and training in IT and computer science received by translators who think they could (5a) or could not (5b) benefit from high quality MT

4.5 Use of MT and employment type

There were six groups of respondents according to the type of organisation they worked for. The two largest groups were freelancers who had an agency but also worked independently apart, and freelancers who only worked independently (45% and 34% respectively). Only 12% of the population worked exclusively with an agency. Other respondents worked as in-house translators in a company (6%). Finally, 2% worked in a government or public institution and 1% of the respondents were students.

Employment type	**Group 1**	**Group 2**	**Group 3**	**Group 4**
Agency and indep.	127	41	44	110
Independent	71	18	38	115
Agency	36	9	11	30
Company	16	5	9	14
Institution	5	1	1	1
Student	3	1	5	1

Table 5: Use of MT by employment type.

Table 5 puts together the summary statistics for the type of employment and the use of MT. Group 1

corresponds to the group “Currently using MT”, Group 2 to “Used MT before”, Group 3 was “Currently no using MT, but might in the future”, Group 4 was “Not using and not planning to”. We can see that Group 4, which never used MT, is largest for independent freelancers, while Group 1 “Currently using MT” is larger for translators who work both independently and with an agency and also for the ones who only work with an agency. A reason for that might be that the translators who collaborate with LSP (Language Service Provider) companies receive more information or training on MT and are in some way encouraged to use it. In addition, many LSPs have their own proprietary engines, which translators have access to, and which are trained for specific domains and language pairs and, therefore, provide better translation quality. The results of the chi-square test confirmed that there is a dependency between the two variables (p-value = 1.475e-14).

It has to be mentioned that, as one can observe in Table 5, groups 1 and 4 are generally larger than groups 2 and 3 almost for all lines. This is due to the total distribution of the participants in these four groups, all types of employment put together (consider Figure 1). Since the total number of translators in groups 1 and 4 is bigger, it is also true for each employment type, i.e. for each line.

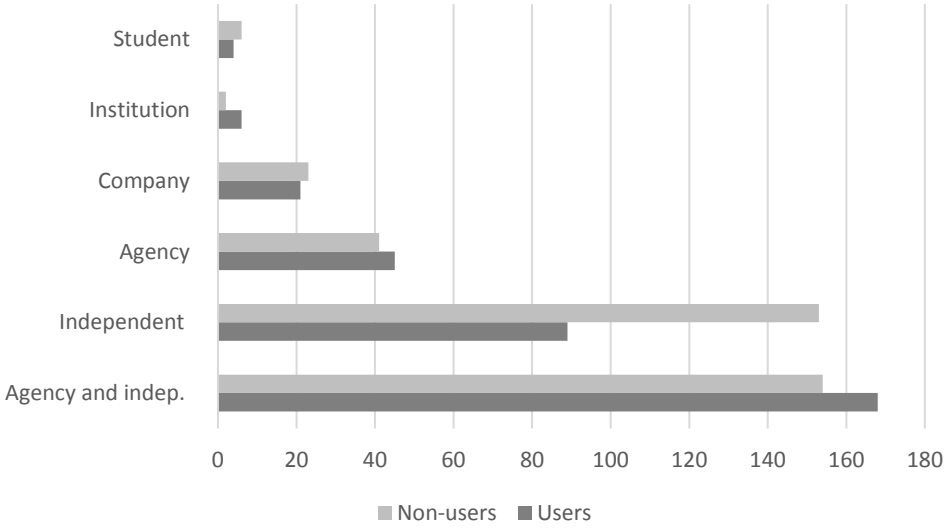


Figure 6: Use of MT by employment type.

Consider Figure 6, which compares only the two groups of users and non-users of MT. We can see that the number of translators who did not use MT is much higher for independent freelancers. Translators who work with agencies are almost e qually likely to use or not to use MT with a slightly higher result for users.

5 Discussion of results and conclusions

In this article we analysed a part of results of a user survey on translation technologies, which concerned various aspects of the use of machine translation by professional translators. The purpose of the analysis was to identify the factors that might influence the usage rate of MT, the general idea of usefulness of this technology and the attitude towards its future advancement.

In general, MT usage rate was higher compared to previous surveys. In addition, most translators saw advancements in MT as a positive process, thinking that it can increase their income and productivity, while leaving more space for creative and challenging parts of translation process. Even though some translators still perceive high-quality MT as a threat to their profession, most of them realised that perfect automatic translation will not be achieved in the near future. Better quality, however, is still absolutely necessary for many users.

Additionally, we investigated what other factors, apart from MT system performance, might be related with MT usage rates. Thus, we considered whether MT is used more with more popular languages for which more parallel data is available, whether is more popular among translators of specific domains,

and whether translators of particular domains can benefit more from high quality machine translation than others. Furthermore, we considered the relation of translators' knowledge and training in IT to their use and attitudes regarding MT, and, finally, different groups according to types of employment.

In contradiction to our initial hypothesis, there has not been identified any dependency between the languages and the MT usage rates. In other words, MT is equally used both with resource-poor and resource-rich languages despite the differences in the quality of the MT output. On the other hand, domain of specialisation seems to be an important factor influencing translators' decision to use MT. Technical domains like information and communications technology (ICT) and computer science imply higher MT usage rates. However, when asked about possible benefits from high quality machine translation, translators of all domains seem to have similar opinions. In other words, MT is currently used with domains where the quality is relatively better, whereas increased translation quality for all domains will lead to higher usage rates. Nevertheless, about a quarter of respondents still do not see any benefit in high quality MT, which is probably due to the fact that they have not found a way to incorporate it to their workflow.

High level of IT competence and courses in IT and CAT tools also seem to increase the usage rates of MT, so knowledge and training in the field and deeper understanding of the technology might help translators make better use of it. And finally, independent freelance translators who do not work in collaboration with an agency are less likely to resort to MT, which might be due to the differences in the project management process and translation workflow which independent translators have. In addition, some agencies have their own proprietary engines trained for specific domain, which can provide more usable output.

Acknowledgements

Anna Zaretskaya is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471. I would like to thank Prof. G. Corpas and Dr. M. Seghiri for their valuable comments and suggestions to improve the paper.

I would also like to thank all the participants who took time to answer our survey, as well as all the commercial partners and universities in the EXPERT project who helped with the survey distribution, and specifically Alessandro Cattelan and Translated.

References

Carl F. Auerbach and Louise B. Silverstein. 2003. *Qualitative Data: An Introduction to Coding and Analysis*. New York University Press.

Tehmina N. Basit. 2003. Manual or electronic? the role of coding in qualitative data analysis. *Educational Research*, 45(2):143–154.

Lynne Bowker and Gloria Corpas Pastor, 2014. *Handbook of Computational Linguistics*, chapter Translation Technology. Oxford University Press.

Stephen Doherty, Federico Gaspari, Declan Groves, Josef van Genabith, Lucia Specia, Aljoscha Burchardt, Arle Lommel, and Hans Uszkoreit. 2013. QTLaunchPad – Mapping the Industry I: Findings on Translation Technologies and Quality Assessment. European Comission Report. Technical report.

Heather Fulford and Joaquin Granell-Zafra. 2005. Translation and technology: a study of uk freelance translators. *The Journal of Specialised Translation (JoSTrans)*, 4:2–7.

GilbaneGroup. 2009. Multilingual product content. Transforming traditional practices to global content value chains. Technical report, The Gilbane Group, June.

Elina Lagoudaki. 2006. Translation memories survey 2006: Users' perceptions around TM use. In *Proceedings of the International Conference Translating & the Computer 28*, pages 15–16. ASLIB.

Elina Lagoudaki. 2008. *Expanding the Possibilities of Translation Memory Systems: From the Translator's Wishlist to the Developer's Design*. Ph.D. thesis, University College London.

Eun Sul Lee and Ronald N. Forthofer. 2006. *Analyzing Complex Survey Data*, volume 71. Sage Publications, second edition.

J. N. K. Rao and Alastair J. Scott. 1981. The analysis of categorical data from complex sample surveys: Chi-squared tests for goodness of fit and independence in two-way tables. *Journal of the American Statistical Association*, 76(374):221–230.

SDL. 2009. Automated Translatoin 2009 Survey. Technical report. `http://www.cmswire.com/cms/enterprise-cms/sdl-survey-says-automated-translation-gains-momentum-007323.php`.

Ruth Torres Domínguez. 2012. The 2012 use of translation technologies survey. `http://mozgorilla.com/download/19/`.

Trad'Online. 2011. Translation business and translators. Technical report, Trad Online.

Anna Zaretskaya, Gloria Corpas Pastor, and Miriam Seghiri. 2015. Translators' requirements for translation technologies: Results of a user survey. In *Proceedings of the AIETI7 Conference New Horizons is Translation and Interpreting Studies*. AIETI.

Statistical Automatic Post Editing

Santanu Pal
Universität des Saarlandes,
Saarbrücken, Germany
santanu.pal@uni-saarland.de

Abstract

The major goal of Automatic post-editing (APE) is to reduce the human post-editing efforts and increase human post-editing productivity by improving the quality of machine translation (MT) output in terms of fluency and adequacy, i.e., the translations produced should be as close as possible to manually post-edited translations. In this paper, we apply Hierarchical Phrase Based Statistical MT (HPB-SMT) to the task of monolingual Statistical APE (SAPE). The SAPE system takes raw Italian Machine Translation output of Google Translator and produces post-edited Italian language text. We carried out the SAPE experiments on a multiway parallel dataset consisting of English Text, Italian MT output and corresponding manually post-edited translations. The evaluation process was carried out into two directions: (i) using automatic MT evaluation metrics (BLEU, TER and METEOR) and (ii) manual evaluation with four professional translators. In both cases, our SAPE system output not only provides better translations than the standard MT output, but also reduces the post-editing efforts in accordance with the translators' perspectives.

1 Introduction

In Machine Translation (MT), the term "Post-Editing" (PE) is defined as the correction by human over the translation produced by an MT system (Veale and Way, 1997), often with minimum amount of manual labor (TAUS Report, 2010), as a process of modification rather than revision (Loffler-Laurian, 1985). Current MT systems fail to deliver perfect translation output. To achieve sufficient quality output, translations often need to be corrected or post-edited by human translators. A major goal of using an APE system is to reduce the effort of the human post-editors or translators by automatically customising the MT output to a particular translation domain.

Recently, Automatic MT post-editing (APE) (Knight and Chander, 1994) has produced more reliable translation compared to raw MT output. The advantage of an APE system is that it can adapt to any Black-box MT engine output as an input and provide possible automatic PE output without the need of having to retrain or re-implement the applied MT engine. In terms of implementation, Phrase Based Statistical Machine Translation (PB-SMT) (Koehn et al., 2003) can be applied as APE system (Simard et al., 2007a).

MT translations generally suffer from a number of adequacy errors which include: incorrect lexical choice, word ordering, word insertion, word deletion, etc. We describe our APE system that covers lexical choice errors, word insertion and deletion between the MT translation and PE translation using the HPB-SMT method. This method is also able to correct word order errors to some extent. The performance of an SMT system heavily relies on bilingual data and word alignment. We propose a hybrid word alignment method which provides well estimated word alignment links for our SAPE system (described in Section 3). During phrase extraction, the system can automatically handle and estimate the following errors:

- Word insertion error (by considering one-to-many alignment links between MT–PE aligned data)
- Word deletion error (by considering many-to-one alignment links between MT–PE aligned data)

- Lexical error (by estimating lexical weighting during model estimation)
- Word ordering (using a hierarchical model facilitates word ordering, because it uses synchronous context free grammar (SCFG) based hierarchical phrases)

The evaluation process has been carried out in two directions: (i) automatic and (ii) human evaluation with 4 expert translators. The automatic evaluation was carried out using three automatic evaluation metrics - BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Lavie and Agarwal, 2007), results suggest that SAPE improves over raw MT output. Evaluation using human judgments also shows that the SAPE improves overall translation adequacy for 30% of the post-edited sentences.

The remainder of the paper is organised as follows. Section 2 gives an account of related research, Section 3 describes the various components of our system, in particular the preprocessing module, hybrid word alignment module and the Hierarchical PB-SMT model. In Section 4, we outline the complete experimental setup. Section 5 presents the results of automatic and human evaluation with some analysis, followed by conclusion and avenues for further research in Section 6.

2 Related Research

Simard et al. (2007a) and Simard et al. (2007b) applied Statistical MT (SMT) for post-editing that handles the repetitive nature of errors typically made by rule-based MT (RBMT) systems. The SMT system was trained on the output of the rule-based system as the source language and reference human translations as the target language. This PB-SMT based APE system was able to correct systematic errors produced by the RBMT system and reduce the post-editing effort.This approach achieved large improvements in performance not only over the baseline rule-based system but also over a similar PB-SMT used in a standalone mode. Denkowski (2015) proposed a method for real time integration of post-edited MT output into the translation model. He extracted a grammar for each input sentence and applied to the model.

Post-editing rule-based MT (RBMT) output can also be done by using statistical information from the trained SMT models (Lagarda et al., 2009).The approach involves training an SMT system and applying the trained models for the purpose of correction to the rule-based output. Rosa et al. (2012) and Mareček et al. (2011) applied a rule-based approach to APE of English-to-Czech MT outputs on the morphological level. They used 20 hand-written rules based on the most frequent errors encountered in translation. The method efficiently corrects morphosyntactic categories of a word such as number, case, gender, person and dependency label.

Various automatic or semi-automatic post-processing techniques to implement corrections for repetitive errors or fully automate PE have been developed, although the resulting MT output still needs to be post-edited by humans in order to produce publishable quality translation (Roturier, 2009; TAUS/CNGL Report, 2010). Even though MT output needs human PE, it is often faster and cheaper to post-edit MT output than to perform human translation from scratch. In some cases, recent studies have even shown that the quality of MT plus PE can exceed the quality of human translation (Fiederer and OBrien, 2009; Koehn, 2009; DePalma and Kelly, 2009) as well as the productivity (Zampieri and Vela, 2014). Aimed at cost-effective and timesaving use of MT, the PE process needs to be further optimised (TAUS/CNGL Report, 2010).

There have also been many studies regarding the impacts of various factors and methods; those were examined against the amount of PE effort. However, those studies have not been conducted to observe PE effort in a commercial work environment. The overall purpose of the present study is to answer two fundamental questions "What would be the optimal design of a PE system?" which is ultimately determined by the quality of MT output. And "How can human involvement be optimised in a PE system to reduce post editing effort?"

3 System Description

The proposed system consist of three basic components: corpus cleaning as well as preprocessing, a hybrid implementation of an improved word alignment model and a HPB-SMT PE system integrated

with hybrid word alignments. The proposed APE system has been trained on monolingual Italian MT output provided by Google Translate (GT) and the manually post-edited output.

3.1 Corpus Cleaning and Preprocessing

The training data used for the experiments was developed within the MateCat[1] project and contains 312K sentences. We utilised all the parallel training data (which includes Europarl as well as News Commentary) for English–MT output as well as the post-edited Italian MT output provided by MateCat. The corpus contains some non-Italian as well as non-English words and sentences. Therefore, we apply the Language Identifier (Shuyo, 2010) on both bilingual English–Italian MT output and MT-output–PE parallel data as well monolingual Italian corpora. We discarded those sentences from the bilingual training data which are considered as belonging to the different language or contained different language segments. The same method has been also applied to the monolingual Italian data. The cleaning process of the training corpus was carried out first by calculating the global mean ratio of the number of characters of source sentence to target sentence and then filter out sentence pairs that exceeds or fall below 20% of the global ratio (Tan and Pal, 2014). We sorted the entire parallel training corpus based on their sentence length and removed duplicates. We apply tokenisation and punctuation normalisation using Moses scripts. In the final steps of cleaning, we filtered the parallel training data on maximum allowable sentence length of 100 and sentence length ratio of 1:2 (either direction).

3.2 Improved Word Alignment

3.2.1 Word Alignment Using GIZA++

GIZA++ (Och and Ney, 2003) is a statistical word alignment tool which implements maximum likelihood estimators for all the IBM 1-5 models and an HMM alignment model as well as Model 6. GIZA++ facilitates fast development of statistical machine translation (SMT) systems. The model parameters of GIZA++ acquire better estimation from very large amount of parallel data. But it is hard to define what would be sufficient amount of parallel data. However, for limited amount of parallel resources, the quality of word alignments is typically quite low and it also deviates from the independence assumptions made by the generative models. GIZA++ has some draw-backs. It allows at most one source word to be aligned with each foreign word. To resolve this issue, some techniques have already been applied such as symmetrisation methods where the parallel corpus is aligned using bidirectional training and then the two alignment tables are reconciled using different heuristics, e.g., union, intersection and most recently grow-diagonal-final and grow-diagonal-final-and heuristics (Koehn, 2010). In spite of these heuristics, the word alignment quality is still low and calls for further improvement. We describe our approach of improving word alignment quality in the following subsections.

3.2.2 Berkley Aligner

Like GIZA++ , the Berkley Aligner is also used to align words across sentence pairs in a bilingual parallel corpus. The Berkeley Aligner (Liang et al., 2006) allows both unsupervised and supervised approaches to align words from parallel corpora. We initially train on the parallel corpus using the fully unsupervised method of producing Berkley word alignments. The Berkeley aligner is an extension of the Cross Expectation Maximization word aligner. The aligner uses agreement between two simple sequence-based models by training and facilitates substantial error reductions over standard models. Moreover, it is jointly trained with HMM models, and as a result AER (Vilar et al., 2006) reduces by 29%. Berkeley Aligner is a very useful word aligner because it allows for supervised training, enabling us to take knowledge from already aligned corpora or we can use the same corpus by updating the alignments using a rule based or a edit distance based alignment system. In this work, the alignment table produced by the unsupervised aligner is corrected using TER word alignment. TER has generated an alignment between two monolingual text (in this case, MT and PE text), which is mostly based on edit distance. This motivated us to use TER alignment as a Gold Standard. Then we apply this corrected alignment table as the gold standard training data for the supervised aligner.

[1] https://www.matecat.com/

3.2.3 SymGiza++

SymGiza++ (Junczys-Dowmunt and Szał, 2012) modifies the counting phase of each model of Giza++ to allow updating of the symmetrisised models between the chosen iterations of the original training algorithms. It computes symmetric word alignment models with the capability of taking advantage of multi-processor systems. Experimental results show that the alignment quality improves by more than 17% compared to Giza++.

3.2.4 TER Alignment

TER (translation edit rate, or translation error rate) (Snover et al., 2006) was originally developed as an automatic MT evaluation metric. TER is an edit distance based metric and measures the ratio between the number of edit operations that are required to turn a hypothesis H (MT output) into a reference R (in this case the PE translation) to the total number of words in R. The allowable edit operations include insertion (Ins), substitution (Sub), deletion (Del) and phrase shifts (Shft). As a byproduct of finding the minimum edit distance, it produces an alignment between the hypothesis and the reference. In the monolingual SAPE task, we make use of TER alignment as a potential alignment between the MT output and the PE translation. TER is computed as in equation (1).

$$TER(H, R) = \frac{(Ins + Del + Sub + Shft) * 100\%}{\textit{total number of words in R}} \quad (1)$$

3.2.5 METEOR Alignment

Like TER, METEOR (Lavie and Agarwal, 2007) is another automatic MT evaluation metric which provides alignment between the hypothesis and reference. Given a pair of strings such as H and R to be compared, METEOR initially establishes a word alignment between them. The alignment is a mapping method between words in H and R, which is built incrementally by the following sequence of word-mapping modules:

- **Exact:** maps if they are exactly the same.
- **Porter stem:** maps if they are the same after they are stemmed using the Porter stemmer.
- **WN synonymy:** maps if they are considered synonyms in WordNet.

If multiple alignments exist, METEOR selects the alignment for which the word order in the two strings is most similar (i.e. having fewest crossing alignment links). The final alignment is produced between H and R as the union of all stage alignments (e.g. Exact, Porter Stem and WN synonymy).

3.2.6 Hybridization

The hybrid word alignment method combines three different kinds of statistical word alignment methods including: Giza++ word alignment with grow-diag-final-and (GDFA) heuristic (Koehn, 2010), Berkeley word alignment and SymGiza++ word alignment as well as two different kind of edit distance based aligners such as TER and METEOR. We have followed two different strategies to combine all word alignment tables (Pal et al., 2013).

Union: In the union method, we hypothesise that all alignments are correct. All the alignment tables are unioned together and duplicate entries are removed.

ADD additional Alignments: This method follows the following heuristic. We consider either of the alignments generated by GDFA (A1), Berkeley aligner (A2), or SymGiza++ (A3) as the standard alignment as TER (A4) and METEOR (A5) fail to align all words in the monolingual Italian MT–PE parallel sentences. The alignment given by A4 and A5 are based on the edit distance method, therefore, it can not align all words between monolingual Italian MT–PE parallel sentences. From the five

alignments A1–A5, we propose the alignment combination method as described in algorithm 1.

ALGORITHM: 1

- Step 1: Choose a standard alignment (S_A) from A_1, A_2 and A_3.
- Step 2: Correct the alignment of S_A by looking at the alignment tables of A4 and A5.
- Step 3: Find additional alignment from other alignments e.g. A_2 , A_3, A_4 and A_5 using intersection method, if A_1 considered as S_A.

$$(A_2 \bigcap A_3 \bigcap A_4 \bigcap A_5) \tag{2}$$

- Step 4: Add additional alignment to S_A.

3.3 HPB-SMT

Hierarchical PB-SMT is based on Synchronous Context Free Grammar (SCFG) (Aho and Ullman, 1969). SCFG rewrites rules on the right-hand side with aligned pairs (Chiang, 2007).

$$X \rightarrow < \gamma, \alpha, \sim > \tag{3}$$

where X represents a nonterminal, γ, α represent sequences of both terminal and nonterminal strings and $\sim$ represents a one-to-one correspondence between occurrences of nonterminals appearing in γ and α.

The weight of each rule is defined as :

$$w(X \rightarrow < \gamma, \alpha, \sim >) = \prod_i \phi_i(X \rightarrow < \gamma, \alpha, \sim >)^{\lambda_i} \tag{4}$$

where ϕ_i are features defined on each rule and λ_i is the weight of ϕ_i. The features are associated with 4 probabilities: frequency probabilities $P(\gamma|\alpha)$, $P(\alpha|\gamma)$, lexical frequency probabilities $P_w(\gamma|\alpha)$, $P_w(\alpha|\gamma)$ and a Phrase penalty $exp(-1)$.

There exist two additional rules called "glue rule" or "glue grammar" :

$$S \rightarrow < SX, SX > \tag{5}$$

$$S \rightarrow < X, X > \tag{6}$$

These rules are used when no rules could match or the span exceeds a certain length (search depth: set the same as the initial phrase length limit). These rules simply monotonically connect translations of two adjacent blocks together.

The weight of the above type of rule is defined as

$$w(S \rightarrow < SX, SX >) = exp(-\lambda_g) \tag{7}$$

where λ_g controls the model's preference for hierarchical phrases over serial combination of phrases.

The weight ($w(d_g)$) of the derivation grammar (d_g) for generated source (f_d) and target (e_d) string, is the product of the weights of the rules used in translation $w(r)$, language model probability P_{lm} and the word penalty $exp(-\lambda_{wp}|e|)$ with some control over the length of the target output (e). The representation of d_g can be defined as a triplet $< r, i, j >$, where, r stands for grammar rule to rewrite a nonterminal that spans $f_{d_i}^{j}$ on the source side.

$$w(d_g) = \prod_{<r,i,j> \in d_g} w(r) \times P_{lm}^{\lambda_{lm}} \times exp(-\lambda_{wp}|e|) \tag{8}$$

4 Experiment

4.1 Data

The dataset that has been used for the experiments is developed in the MateCat project. The data consist of 312K parallel sentences. The parallel data (in this case, MT output as the source language and reference human translations as the target language) are cleaned and processed by using our preprocessing module (see Section 3.1). The provided data was noisy. After cleaning, we obtained a 213,795 sentence-aligned MT–PE parallel corpus from a mixed domain (Europarl and News commentary) for our present experiment. We randomly extracted 1000 sentences each for the development set and test set from the initial parallel corpus, and treated the rest (211,795) as the training corpus. The Italian monolingual corpus for language modelling has been prepared by using Italian PE data in addition with the Europarl monolingual clean corpus. The monolingual corpus consists of 49,483,285 words.

4.2 Experimental settings

We used the Hierarchical PB-SMT model. For building our APE system, we experimented with various maximum phrase lengths for the translation model and n–gram settings for the language model. We found that using a maximum phrase length of 7 and a 5-gram language model produced best results in terms of BLEU scores (Papineni et al., 2002). We performed smoothing using the Good-Turing technique (Foster et al., 2006).

The other experimental settings our improved hybrid word alignment models (cf. Section 3.2) integrated with in the Hierarchical phrase-extraction (Chiang, 2005). The 5-gram target language model was trained using KenLM (Heafield, 2011). The system tuning was carried out using both k-best MIRA (Cherry and Foster, 2012) and Minimum Error Rate Training (MERT) (Och, 2003) on the held-out development set (devset). After the parameters were tuned, decoding was carried out on the held out test set.

5 Evaluation

The evaluation process was carried out into two ways: (i) Automatic evaluation and (ii) Human evaluation by 4 expert translators on the 1000 sentences of the test set automatically post edited by the proposed system. Out of 1000 sentences, 145 sentences are different from the raw Google Translate translations output i.e. 14.5% sentences are post edited by the system, rest of sentences remain similar and are not affected by APE. We analyzed those 145 sentences to find out whether they really improve the quality or not.

5.1 Automatic Evaluation

We evaluated MT quality using three well known automatic MT evaluation metrics: BLEU, METEOR and TER. We also performed sentence level BLEU evaluation. Based on sentence level BLEU score, the evaluation results in Table 1 shows that 91 out of 145 sentences provided by Automatic Post Editing (APE) system are better in quality than Google Translation (GT). However, for the rest (54) of the sentences, quality is degraded by APE; for these 54 sentences the Google Translation is of better quality than the APE output. To some extent this may be PE reference sentences are biased towards GT output. However, manual inspection showed that some of those (31) sentences are worse than the GT output.

Metric	APE better	GT better	Tie	% Gain	% Loss
Sentence BLEU	91	54	855	9.1%	5.4%

Table 1: Automatic sentence level evaluation.

Table 2 provides a systematic comparison between the GT and APE systems with three different evaluation metrics. In all cases, our proposed system performed better. Table 2 also shows the relative

improvement over GT is maximal with respect to TER. The improvements vary with different metrics. This motivated us to perform manual human evaluation by professional translators.

Metric	APE	GT	% Relative Improvements
BLEU	63.87	61.26	4.2%
TER	28.67	30.94	7.9%
METEOR	73.63	72.73	1.2%

Table 2: Automatic evaluation over 1000 sentences.

5.2 Human Evaluation

For human evaluation, we developed a polling scheme with three different options for a particular source English segment. Translators act as voters and make a choice between options as to which of the outputs looks better to them. The first two of the three options are for Italian translations provided by APE and GT and the third option is *uncertain* (U) which is applicable whenever the translators are uncertain about which translation is better i.e. both the GT and APE translations are equally good or worse to them. To avoid biasses of the translators to a particular system, we randomly swap the position of APE and GT translations so that the translators do not know which system they are contributing their votes to; they just choose the best of the 2 translations. The human evaluation involved four professional translators. All these 4 translators are working for Translated srl[2], Rome, Italy. The profile of the translators are presented in Table 3.

Table 3 shows the results of the pole scheme (by human evaluation) of the raw GT output compared to the automatic post-edited (APE) output. The values in the table represent how many translations have been chosen by each translator for each particular system. The evaluation of the polling process was carried out with 145 translations. We conducted the voting process serially to avoid any conflict between the translators. From Table 3, we can easily conclude that all translators showed the tendency of adopting to the new system and they accept the APE system to their regular work. The winning APE system received 7% vote compared to 2.5% vote by the GT system, while 85% votes are tied as both the APE and the GT translations are the same for these sentences and 5% are neutral as the translators were undecided for those sentences. As each translators received the same 145 sentences for judgment, the total vote received by APE system is 105, GT: 62 and Uncertain: 94. After deep analysis, we found that all the 4 translators agreed on 27 translations provided by the APE system, 6 translation provided by GT and 9 translation are neutral. Therefore, our APE system is able to improve the quality approximately by 3% over 1000 sentences with a 0.6% decrease and 0.9% neutral.

6 Conclusions and Future Work

The proposed APE system was successful in improving over the baseline MT system performance. Although some APE translations were deemed worse than the original MT output by the human evaluators, however, they were very few in numbers. Manual inspection revealed that these lower quality APE translations are very similar to the original MT translations. These worse translations can be avoided by adding more features (e.g., syntactic or semantic) which can also improve the overall performance of the post-editing system. The presented system can easily be plugged into any state-of-the-art system and the runtime complexity is similar to that of other statistical MT systems. In future, we will try bootstrapping strategies for further tuning the model and add more sophisticated features beyond the lexical level. We will improve our hybrid word alignment algorithm by incorporating additional word aligners such as fastaligner, Anymaligner, etc. We also want to extend the system by incorporating source knowledge as well as improving word ordering by using the Kendall reordering method. To consolidate

[2]http://www.translated.net/en/

	Degree	Expertise	Experience in years	APE	GT	U
Translator 1	Translation	EN,FR → IT	1	91	22	32
Translator 2	Linguistic and Cultural Studies	EN,FR, ES, CA → IT	2	57	17	71
Translator 3	European Languages and Cultures	EN, FR, ES, DE → IT	1	72	37	36
Translator 4	Business & Administration	EN → IT	1	65	23	58

Table 3: Manual Evaluation with 4 expert translators for 145 sentences, (EN=English, DE= German, FR= French, ES = Spanish, CA= Catalan, IT= Italian).

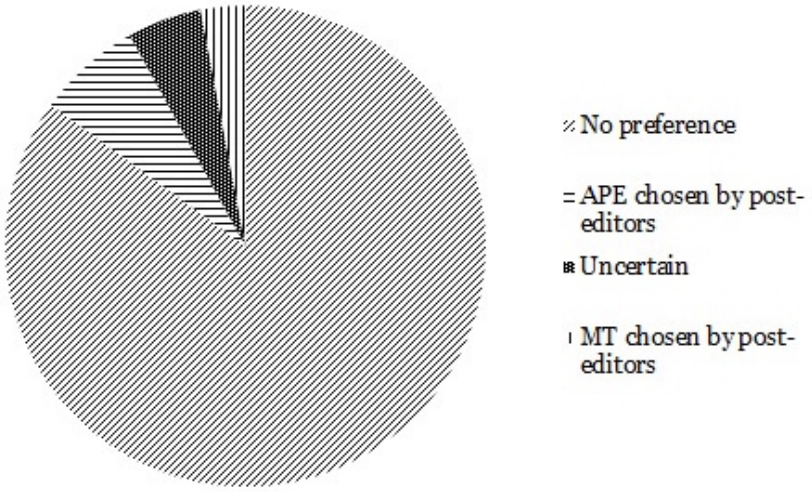


Figure 1: Overall evaluation by human.

the user evaluation, we will measure inter-annotator agreement. We will also evaluate our system in a real-life setting in commercial environment to analyse time gain and productivity gain provided by automatic post-editing.

Acknowledgements

The research leading to these results has received funding from the EU FP7 Project EXPERT the People Programme (Marie Curie Actions) under REA grant agreement n° 317471.

References

Alfred. V. Aho and Jeffrey. Ullman. 1969. Translations on a Context Free Grammar. In *Proceedings of the First Annual ACM Symposium on Theory of Computing*, pages 93–112.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436.

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Michael Denkowski. 2015. *Machine Translation for Human Translators*. Ph.D. thesis, Carnegie Mellon University.

Donald A. DePalma and Nataly Kelly. 2009. Project Management for Crowdsourced Translation: How User-Translated Content Projects Work in Real Life. *Translation and Localization Project Management: The Art of the Possible*, pages 379–408.

Rebecca Fiederer and Sharon OBrien. 2009. Quality and Machine Translation: a Realistic Objective. *Journal of Specialised Translation*, 11:52–74.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable Smoothing for Statistical Machine Translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Marcin Junczys-Dowmunt and Arkadiusz Szał. 2012. SyMGiza++: Symmetrized Word Alignment Models for Statistical Machine Translation. In *Proceedings of the 2011 International Conference on Security and Intelligent Information Systems*, pages 379–390.

Kevin Knight and Ishwar Chander. 1994. Automated Postediting of Documents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, pages 779–784.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Philipp Koehn. 2009. A Process Study of Computer-aided Translation. *Machine Translation*, 23(4):241–263.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Antonio Lagarda, Vicent Alabau, Francisco Casacuberta, Roberto Silva, and Enrique Díaz-de Liaño. 2009. Statistical Post-editing of a Rule-based Machine Translation System. In *Proceedings of Human Language Technologies*, pages 217–220, Stroudsburg, PA, USA.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 104–111.

Anne-Marie Loffler-Laurian. 1985. Traduction automatique et style. *Babel*, 31(2):70–76.

David Mareček, Rudolf Rosa, Petra Galuščáková, and Ondřej Bojar. 2011. Two-step Translation with Grammatical Post-processing. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 426–432.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167.

Santanu Pal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2013. A Hybrid Word Alignment Model for Phrase-Based Statistical Ma-chine Translation. *ACL 2013*, pages 94–101.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318.

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Stroudsburg, PA, USA.

Johann Roturier. 2009. Deploying Novel MT technology to Raise the Bar for Quality: a Review of Key Advantages and Challenges. In *Proceedings of the twelfth Machine Translation Summit*.

Nakatani Shuyo. 2010. Language Detection Library for Java.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical Phrase-based Post-editing. In *In Proceedings of NAACL*.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Liling Tan and Santanu Pal. 2014. Manawi: Using Multi-word Expressions and Named Entities to Improve Machine Translation. In *Proceedings of Ninth Workshop on Statistical Machine Translation*.

TAUS Report. 2010. Post editing in practice. Technical report, TAUS.

TAUS/CNGL Report. 2010. Maschine Translation Post-Editing Guidelines Published. Technical report, TAUS.

Tony Veale and Andy Way. 1997. Gaijin: A Bootstrapping, Template-driven Approach to Example-based MT. In *Proceedings of the Recent Advances in Natural Language Processing*.

David Vilar, Maja Popovi, and Hermann Ney. 2006. AER: Do we need to improve our alignments. In *In Proceedings of the International Workshop on Spoken Language Translation*, pages 205–212.

Marcos Zampieri and Mihaela Vela. 2014. Quantifying the Influence of MT Output in the Translators Performance: A Case Study in Technical Translation. In *Proceedings of the EACL Workshop on Humans and Computer-assisted Translation (HaCat)*, May.

Assessing Comparable Corpora through Distributional Similarity Measures

Hernani Costa
LEXYTRAD, University of Malaga, Spain
hercos@uma.es

Abstract

Describing, comparing and evaluating corpora are key issues in corpus-based translation and corpus linguistics for which there is still a notable lack of standards. Bearing this in mind, this paper aims at investigating the use of textual distributional similarity measures in the context of comparable corpora. More precisely, we address the issue of measuring the relatedness between documents by extracting and measuring their common content. For this purpose, we designed and applied a methodology that exploits available natural language processing technology with statistical methods. Our findings showed that using a list of common entities and a simple, yet robust and high performance set of distributional similarity measures was enough to describe and assess the degree of relatedness between the documents in a comparable corpus.

1 Introduction

The use of comparable corpora has been considered an essential resource in several research domains such as Natural Language Processing (NLP), terminology, language teaching, and automatic and assisted translation, amongst others. Nevertheless, an inherent problem to those who deal with comparable corpora in a daily basis is the uncertainty about the data they are dealing with. Indeed, little work has been done on automatically characterising such linguistic resources and attempting a meaningful description of their content is often a perilous task (Corpas Pastor and Seghiri, 2009). Usually, a corpus is given a short description such as "casual speech transcripts" or "tourism specialised comparable corpus". However, such tags will be of little use to those users seeking for a representative and/or high quality domain-specific corpora. Apart from the usual description that comes along with the corpus, like number of documents, tokens, types, source(s), creation date, policies of usage, etc., nothing is said about how similar the documents are. As a result, most of the resources at our disposal are built and shared without deep analysis of their content, and those who use them blindly trust on the people's or research group's name behind their compilation process, without knowing nothing about the relatedness quality of the corpus.

Bearing this in mind, in this work we try to fill this void by taking advantage of several textual distributional similarity measures presented in the literature. First, we selected a specialised corpus about tourism and beauty domain that was manually compiled by researchers in the area of translation and interpreting studies. Then, we designed and applied a methodology that exploits available NLP technology with statistical methods to assess how the documents correlate with each other in the corpus. Our assumption is that the amount of information contained in a document can be evaluated via summing the amount of information contained in the member words. For this purpose, a list of common entities was used as a unit of measurement capable of identifying the amount of information shared between the documents. Our assumption is that this approach will allow us not only to compute the relatedness between documents, but also to describe and characterise the corpus itself.

The remainder of the paper is structured as follows. Section 2 introduces some fundamental concepts related to distributional similarity measures, i.e. explains the theoretical foundations, related work and the distributional similarity exploited in this experiment. Then, Section 3 presents the corpus used in this work. After applying the methodology described in Section 4, Section 5 presents and discusses the

obtained results in detail. Finally, Section 6 presents the final remarks and highlights our future plans for this work.

2 Distributional Similarity Measures

Information Retrieval (IR) (Singhal, 2001) is the task of locating specific information within a collection of documents or other natural language resources according to some request. In this field, we can find a large number of statistical methods based on words and their (co-)occurrence. Essentially, it involves finding the most frequently used words and treating the rate of usage of each word in a given text as a quantitative attribute. Then, these words serve as features for a given statistical method. Following Harris' distributional hypothesis (Harris, 1970), which assumes that similar words tend to occur in similar contexts, these statistical methods are suitable, for instance to find similar sentences based on the words they contain (Costa et al., 2015a) and automatically extract or validate semantic entities from corpora (Costa et al., 2010; Costa, 2010; Costa et al., 2011). To this end, it is assumed that the amount of information contained in a document could be evaluated by summing the amount of information contained in the document words. And, the amount of information conveyed by a word can be represented by means of the weight assigned to it (Salton and Buckley, 1988). Accordingly, we took advantage of two IR measures commonly used in the literature, the Spearman's Rank Correlation Coefficient (SCC) and the Chi-Square (χ^2) to compute the similarity between two documents written in the same language (see section 2.1 and 2.2). Both measures are particularly useful for this task because they are independent of text size (mostly because both use a list of the common entities), and they are language-independent.

The Spearman's Rank Correlation Coefficient (SCC) distributional measure has been shown effective on determining similarity between sentences, documents and even on corpora of varying sizes (Kilgarriff, 2001; Costa et al., 2015a). It is particularly useful, for instance to measure the textual similarity between two documents because it is easy to compute and is independent of text size as it can directly compare ranked lists for large and small texts.

The χ^2 similarity measure has also shown its robustness and high performance. By way of example, χ^2 have been used to analyse the conversation component of the British National Corpus (Rayson et al., 1997), to compare corpora (Kilgarriff, 2001), and to identify topic related clusters in imperfect transcribed documents (Ibrahimov et al., 2002). It is a simple statistic measure that permits to assess if relationships between two variables in a sample are due to chance or the relationship is systematic.

For all these reasons, distributional similarity measures in general and SCC and χ^2 in particular have a wide range of applicabilities (cf. Kilgarriff (2001) and Costa et al. (2015a)). Indeed, this work aims at proving that these simple, yet robust and high-performance measures allow to describe the relatedness between documents in specialised corpora.

2.1 Spearman's Rank Correlation Coefficient (SCC)

In this work, the SCC is adopted and calculated as in Kilgarriff (2001). Firstly, a list of the common entities[1] L between two documents d_l and d_m is compiled, where $L_{d_l,d_m} \subseteq (d_l \cap d_m)$. It is possible to use the top n most common entities or all common entities between two documents, where n corresponds to the total number of common entities considered $|L|$, i.e. $\{n|n \in \mathbb{N}^0, n \leq |L|\}$ – in this work we use all the common words for each document pair, i.e. $n = |L|$. Then, for each document the list of common entities (e.g. L_{d_l} and L_{d_m}) is ranked by frequency in an ascending order ($R_{L_{d_l}}$ and $R_{L_{d_m}}$), where the entity with lowest frequency receives the numerical raking position 1 and the entity with highest frequency receives the numerical raking position n. In the case of ties in rank, where more than one entity in a document occurs with the same frequency, the average of the ranks is assigned to the tying entities. For instance, if the entities e_a, e_b and e_c had the same frequency and ranked in the 6^{th}, 7^{th} and 8^{th} position, all three entities would be assigned the same rank of $\frac{6+7+8}{3} = 7$. Finally, for each common entity $\{e_1, ..., e_n\} \in L$, the difference in the rank orders for the entity in each document is computed,

[1] In this work, the term 'entity' refers to "single words", which can be a token, a lemma or a stemm.

and then normalised as a sum of the square of these differences $\left(\sum_{i=1}^{n} s_i^2\right)$. The final SCC equation is presented in expression 1, where $\{SCC|SCC \in \mathbb{R}, -1 \geq SCC \leq 1\}$.

By a way of example let e_x be a common entity (i.e. $\{e_x\} \in L$) and $R_{L_{d_l}} = \{1\#e_{n_{d_l}}, 2\#e_{n-1_{d_l}}, ..., n\#e_{1_{d_l}}\}$ and $R_{L_{d_m}} = \{1\#e_{n_{d_m}}, 2\#e_{n-1_{d_m}}, ..., n\#e_{1_{d_m}}\}$ the resulting ranked list of common words for d_l and d_m, respectively. Supposing that e_x is the $3\#e_{n-2_{d_l}}$ and $1\#e_{n_{d_m}}$, i.e. e_x is in the 3^{rd} position in $R_{L_{d_l}}$ and in the 1^{st} position in $R_{L_{d_m}}$, s would be computed as $s_{e_x}^2 = (3-1)^2$ and the result would be 4. Then, this process is repeated for the remain $n-1$ entities and the resulted SCC score will be seen as the similarity value between d_l and d_m.

$$SCC(d_i, d_j) = 1 - \frac{6 * \sum_{i=1}^{n} s_i^2}{n^3 - n} \tag{1}$$

2.2 Chi-Square (χ^2)

The Chi-square (χ^2) measure also uses a list of common words (L). Similarly to SCC, it is also possible to use the top n most common entities or all common entities between two documents, and again in this work we use all the common words for each document pair, i.e. $n = |L|$. The number of occurrences of a common words in L that would be expected in each document is calculated from the frequency lists. If the size of the document d_l and d_m are N_l and N_m and the entity e_i has the following observed frequencies $O(e_i, d_l)$ and $O(e_i, d_m)$, then the expected values are $e_{i_{d_l}} = \frac{N_l * (O(e_i,d_l)+O(e_i,d_m))}{N_l+N_m}$ and $e_{i_{d_m}} = \frac{N_m * (O(e_i,d_l)+O(e_i,d_m))}{N_l+N_m}$. Equation 2 presents the χ^2 formula, where O is the observed frequency and E the expected frequency. The resulted χ^2 score should be interpreted as the interdocument distance between two documents. It is also important to mention that $\{\chi^2|\chi^2 \in \mathbb{R}, 1 \geq \chi^2 < \infty\}$, which means that as more unrelated the common words in L are, the lower the χ^2 score will be.

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{2}$$

Suppose that we have two common entities e_i and e_j between two documents d_l and d_m (i.e. $L = \{e_i, e_j\}$). Table 1 shows a contingency table example. This table contains: i) the observed frequencies (O); ii) the totals in the margins; iii) and the expected frequencies (E), which are obtained by applying the following formula: $\frac{column\ total}{N} * row\_total$, e.g. $E(e_i, d_l) = \frac{14}{26} * 15 = 8.08$. After writing down the expected frequencies in the table, we are ready to calculate the χ^2 score (see Equation 3).

	d_l	d_m	*Total*
e_i	*O*=11 *E*=8.08	*O*=4 *E*=6.92	15
e_j	*O*=3 *E*=5.92	*O*=8 *E*=5.08	11
Total	14	12	26

Table 1: Example of a contigency table.

$$\chi^2 = \frac{(11-8.08)^2}{8.08} + \frac{(3-5.92)^2}{5.92} + \frac{(4-6.92)^2}{6.92} + \frac{(8-5.08)^2}{5.08} = 5.41 \tag{3}$$

3 The INTELITERM Corpus

The INTELITERM[2] corpus is a comparable corpus composed of documents collected from the Internet. Designed to be a specialised comparable corpus, this corpus was manually compiled by researchers

[2] http://www.lexytrad.es/proyectos.html

with the purpose of building a representative corpus for the Tourism and Beauty domain. It contains documents in four different languages (English, Spanish, German and Italian). Some of the texts are translations of each other, yet the majority is composed of original texts. The INTELITERM comparable corpus is composed of several subcorpora, divided by the language and further for each language there are translated and original texts (which will be hereafter referred as language_totd and language_to, respectively). In this work, we used half of the corpus, i.e. all the original and translated documents in English and Spanish (en_to, en_totd, es_to and es_totd, respectively). All the information about these subcorpora is presented in Table 2. In detail, this table shows: the number of documents (nDocs); the number of types (types); the number of tokens (tokens); and the ratio of types per tokens ($\frac{types}{tokens}$) per subcorpus. These values were obtained using the corpus analysis toolkit for concordancing and text analysis software Antconc 3.4.3 (Anthony, 2014).

	nDocs	types	tokens	$\frac{types}{tokens}$	description
en_to	151	11,6k	508,9k	0.023	original
en_totd	61	6,9k	88,5k	0.078	translated
es_to	225	12,6k	253,4k	0.049	original
es_totd	27	3,4k	19,7k	0.174	translated

Table 2: Statistical information about the various subcorpus.

4 Methodology

This section not only describes the methodology used to calculate the similarity between documents using Distributional Similarity Measures (DSMs), but also presents all the tools, libraries and frameworks employed by our system to perform this experiment.

1) **Data Preprocessing**: firstly all the documents within the corpus were processed with the OpenNLP[3] Sentence Detector and Tokeniser. Then, the annotation process was done with the TT4J[4] library, which is a Java wrapper around the popular TreeTagger (Schmid, 1995) – a tool specifically designed to annotate text with part-of-speech and lemma information. Regarding the stemming, we used the Porter stemmer algorithm provided by the Snowball[5] library. A method to remove punctuation and special characters within the words was also implemented. Finally, in order to get rid of the noise, a stopword list[6] was compiled to filter out the most frequent words in the corpus. Once a document is computed and the sentences are tokenised, lemmatised and stemmed, our system creates a new output file with all this new information, i.e. the new document contains: the original, the tokenised, the lemmatised and the stemmed text. Using the stopword list mentioned above a Boolean vector describing if the entity is a stopword or not is also added. This way, the system will be able to use only the tokens, lemmas and stems that are not stopwords.

2) **Identifying the list of common entities between documents**: in order to identify a list of common entities (L), a co-occurrence matrix was built for each pair of documents. Only those that have at least one occurrence in both documents are considered. As required by the DSMs (see section 2), their frequency in both documents is also stored within this matrix ($L_{d_l,d_m} = \{e_i, (f(e_i,d_l), f(e_i,d_m)); e_j, (f(e_j,d_l), f(e_j,d_m)); ...; e_n, (f(e_n,d_l), f(e_n,d_m))\}$). With the purpose of analysing and comparing the performance of different DSMs, three different lists were created to be used as input features: the first one using common tokens, another using common lemmas and the third one using common stems.

3) **Computing the similarity between documents**: the similarity between documents was calculated by applying three different DSMs ($DSMs = \{DSM_{NCE}, DSM_{SCC}, DSM_{\chi^2}\}$, where $_{NCE}$, $_{SCC}$ and

[3] https://opennlp.apache.org
[4] http://reckart.github.io/tt4j/
[5] http://snowball.tartarus.org
[6] Freely available to download through the following URL https://github.com/hpcosta/stopwords.

χ^2 means Number of Common Entities, Spearman's Rank Correlation Coefficient and Chi-Square, respectively), each one calculated using three different input features (list of common tokens, lemmas and stems).

4) **Computing the document final score**: the document final score $DSM(d_l)$ is the mean of the similarity scores of the document with all the documents in the collection of documents, i.e. $DSM(d_l) = \frac{\sum_{i=1}^{n-1} DSM_i(d_l,d_i)}{n-1}$, where n corresponds to the total number of documents in the collection and $DSM_i(d_l, d_i)$ the resulted similarity score between the document d_l with all the documents in the collection.

5 Results and Analysis

In order to describe the corpus in hand, we applied three different Distributional Similarity Measures (DSMs): the Number of Common Entities (NCE), the Spearman's Rank Correlation Coefficient (SCC) and the Chi-Square (χ^2). As a input feature to the DSMs, three different types of entities (tokens, lemmas and stems) were used. Table 3 shows the Number of Common Tokens (NCT) between document on average (av), the SCC and the χ^2 scores along with the associated standard deviations (σ) per measure and subcorpus. Figure 1 presents the resulted average scores per document in a box plot format for all the combinations DSM vs. feature. Each box plot displays the full range of variation (from min to max), the likely range of variation (the interquartile range), the median, and the high maximums and low minimums (also know as outliers). It is important to mention that for this experiment we did not use a sample, but instead the entire corpus in its original size and form, which means that all obtained results and made observations came from the entire population, in this case the various INTELITERM English (en_to and en_totd) and Spanish (es_to and es_totd) subcorpora.

		NCT	SCC	χ^2
en_to	av	163.70	0.42	279.39
	σ	83.87	0.05	177.45
en_totd	av	67.54	0.39	90.38
	σ	35.35	0.05	53.25
es_to	av	31.97	0.41	40.92
	σ	23.48	0.07	38.21
es_totd	av	17.93	0.63	13.40
	σ	8.46	0.14	18.95

Table 3: Average and standard deviation of common tokens scores between document per subcorpus.

The first observation we can make from Figure 1 is that the distributions between the features are quite similar (see for instance Figures 1a, 1d and 1g). This means that it is possible to achieve acceptable results only using raw words (i.e. tokens). Stems and lemmas require more processing power and time to be used as features – especially lemmas due to the Part-of-Speech (POS) tagger dependency and time consuming process implied. In general, we can say that the scores for each subcorpus is symmetric (roughly the same on each side when cut down the middle), which means that the data is normally distributed. There are some exceptions such as the SCC and χ^2 average scores for the es_totd and for the en_to, respectively, which we will discuss later in this section. Another interesting observation is related with the high NCE (see Table 3 and Figures 1a, 1d and 1g) in original documents (en_to and es_to) when compared with documents translated from other languages (en_totd and es_totd, respectively). For example, the subcorpus en_to (which contains original documents) has 163.70 common tokens per document on average (av) with a standard deviation (σ) of 83.87 and the subcorpus en_totd (which contains translated documents) only has 67.54 common tokens per document on average with a σ=35.35 (Table 3). The same observation can be made between the es_to and the es_totd subcorpus (see Figure 1a and Table 3). This fact could happen because these documents are collections of translated documents collected from the Internet, and thus translated from different translator, which implies that different translators use different vocabulary and consequently lower the NCE between the documents will be.

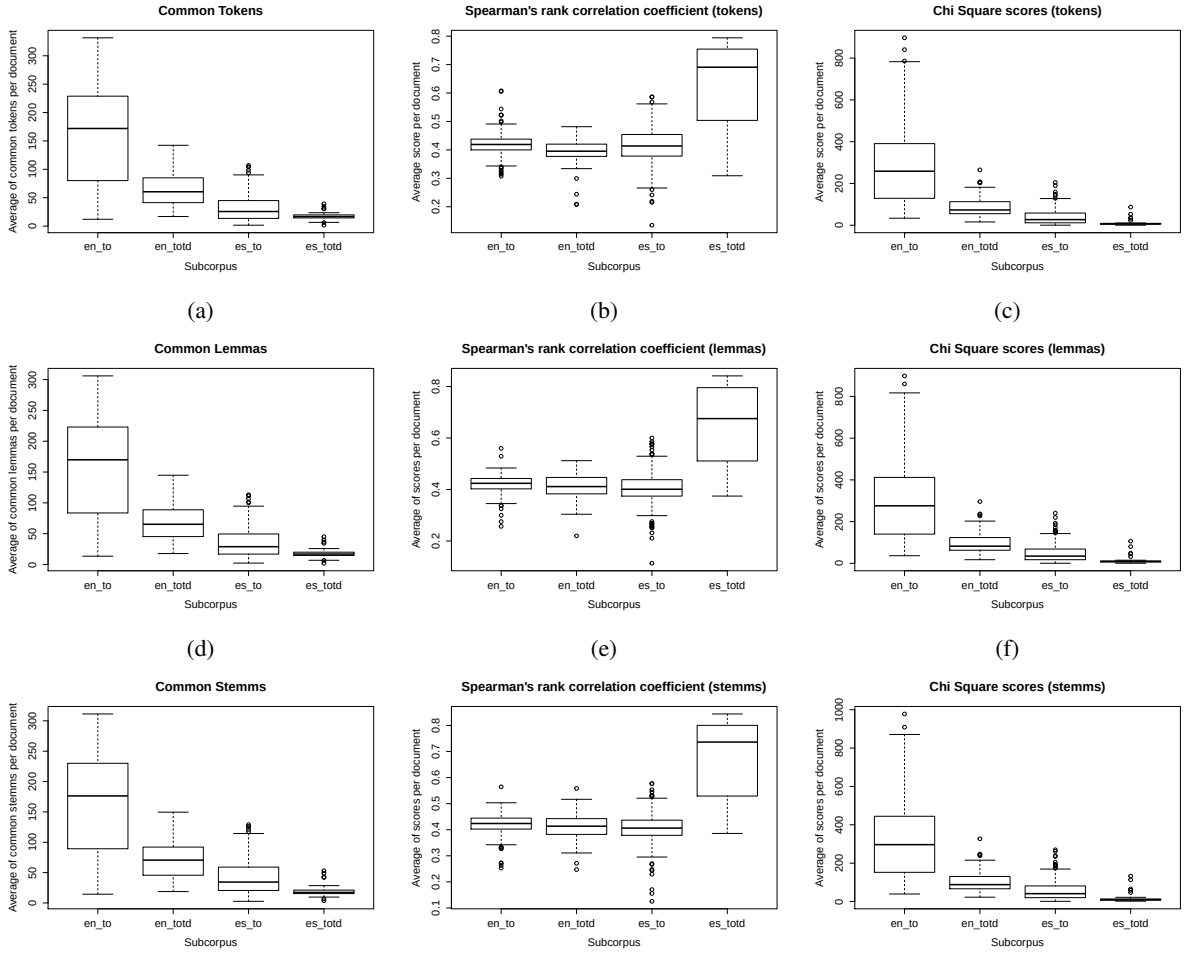


Figure 1: INTELITERM Subcorpus: average scores per document.

Although the Number of Common Tokens (NCT) per document on average is higher for the en_to subcorpus, the interquartile range (IQR) is larger than for the other subcorpora (see Table 3 and Figure 1a), which means that the middle 50% of the data is more distributed and thus the average of NCT per document is more variable. Moreover, longest whiskers (the lines extending vertically from the box) in Figure 1a also indicates variability outside the upper and lower quartiles. Therefore, we can say that en_to has a wide type of documents and consequently some of them are only roughly correlated to the rest of the subcorpus. Nevertheless, the data is skewed right, which means that the majority is strongly similar, i.e. the documents have a high degree of relatedness between each other. This idea can be sustained by the positive average SCC scores presented in Figure 1b and the set of outliers found above the upper whisker. Moreover, the average of 0.42 SCC score and σ=0.05 also implies a strong correlation between the documents in the en_to subcorpus. Likewise, the longest whisker outside the upper quartile and the skewed left χ^2 scores also indicate relatedness between the documents.

Regarding the en_totd subcorpus, the NCT, the SCC and the χ^2 scores (Figures 1a, 1b and 1c) and the average of 90.38 common tokens per document and σ=53.25 (Table 3) suggest that the data is either normally distributed (Figure 1b) or skewed left (Figures 1a and 1c). Considering this results, we can conclude that the documents are highly related.

From all the subcorpora, the es_to subcorpus is the biggest one with 225 documents, 12606 types, 253412 tokens (Table 2). Nevertheless, Table 3 and Figure 1a reveal a lower NCT compared with en_to and the en_totd subcorpora. A theoretical explanation for this phenomenon is that Spanish has richer morphology compared to English. Therefore, due to bigger number of inflection forms per lemma, there

is a larger number of tokens and consequently less common tokens per document in Spanish. When analysing Figures 1a and 1c, the box plots for the es_to subcorpus look similar to the en_totd when shifted up. Except for the longest whisker observed in Figure 1b, the SCC scores also show similar distributions, averages and standard deviations (see Table 3).

As we can see in Figures 1a, 1b and 1c, the average scores per document for es_totd are slightly different from the other box plots. Apart from the low NCT per document, the χ^2 standard deviation higher than its average (18.95 and 13.40, respectively), the SCC variability inside and outside the IQR indicates some inconsistency in the data. This instability can be explained by the subcorpus size, i.e. the small number of documents (27) and by the the low number of types and tokens (3433 and 19736, respectively) and its 0.174 $\frac{types}{tokens}$ ratio. As mentioned by Baker (2006:52), the $\frac{types}{tokens}$ ratio tends to be useful when looking at relatively small documents, and in this specific case this subcorpus only has on average 731 tokens ($\frac{19736}{27} \approx 731$) and 127 types per document ($\frac{3433}{27} \approx 127$), which makes it an excellent test case. When compared with the low ratios from the other subcorpora (see Table 2), – even for this specilised subcorpus – this one can be considered high. If by on one hand, a low ratio can indicate a great number of repetitions (the same word occurring again and again) likely indicating a relatively narrow range of subjects. On the other hand, a high ratio suggests that a more diverse form of language is employed, which can also explain the low NCT and χ^2 scores for this subcorpus in hand. Despite the high SCC, the data is asymmetric and variable (large IQR). This happens because most of the common entities have a low frequency in the documents and consequently they will rank close together in the ranking lists, which results in high SCC scores mostly because of the resulted high value in the numerator (see Equation 1).

To sum up, we can state from the statistical and theoretical evidences that the en_to, the en_totd and the es_to subcorpora look like they assemble highly correlated documents. We can not say the same for the es_totd subcorpus. Due to the small number of documents and scarceness of evidences we can only not reject the idea that this subcorpus is composed of similar documents.

6 Conclusions and Future Work

In this paper we presented and studied various Distributional Similarity Measures (DSMs) for the purpose of describing specialised comparable corpora. As input for these DSMs, we used three different features (lists of common tokens, lemmas and stems). In the end, we conclude that for the data in hand these features had similar performance for all the tested DSMs. In fact, our findings show that instead of using common lemmas or stems, which require external libraries, processing power and time, a simple list of common tokens was enough to describe our data. Moreover, we proved that the corpus used in this experiment is composed of highly correlated documents. The high number of entities shared by its documents, the positive average scores obtained with the SCC measure and their χ^2 scores sustain our claim.

In the immediate future, we intend not only to perform more experiments with these DSMs by adding noisy documents (i.e. out of topic documents) to the corpus and analyse the DSMs performance, but also merge the translated documents from other languages with original ones and prove that translated documents decrease the general relatedness score. Moreover, it is our intention to do the same experiment with other languages, like Italian and German. Apart from that, we also want to test other DSMs, such as Jaccard, Lin and PMI and compare their performance.

Furthermore, these DSMs can be seen as a suitable tool to rank documents by their similarities, which we believe that will be a handy feature to those who manually or semi-automatically compile corpora mined from the Internet. It will allow them to filter out documents with a low level of relatedness when compared with the rest of the documents in the corpus. Indeed, it is our intention to integrate this methodology in the iCorpora application, an ongoing project that aims to design and develop a robust and agile web-based application capable of semi-automatically compile multilingual comparable and parallel corpora (Costa et al., 2014; Costa et al., 2015c; Costa et al., 2015b).

Acknowledgements

Hernani Costa is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement nº 317471. Also, the research reported in this work has been partially carried out in the framework of the Educational Innovation Project TRADICOR (PIE 13-054, 2014-2015); the R&D project INTELITERM (ref. nº FFI2012-38881, 2012-2015); and the R&D Project for Excellence TERMITUR (ref. nº HUM2754, 2014-2017). I would like to thank Prof. Gloria Corpas Pastor, Prof. Ruslan Mitkov and Dr. Miriam Seghiri for their valuable comments and suggestions to improve the paper.

References

Laurence Anthony. 2014. AntConc (Version 3.4.3) Machintosh OS X. Waseda University. Tokyo, Japan. Available from `http://www.laurenceanthony.net`.

Paul Baker. 2006. *Using Corpora in Discourse Analysis*. Bloomsbury Discourse. Bloomsbury Academic.

Gloria Corpas Pastor and Míriam Seghiri. 2009. Virtual Corpora as Documentation Resources: Translating Travel Insurance Documents (English-Spanish). In A. Beeby, P.R. Inés, and P. Sánchez-Gijón, editors, *Corpus Use and Translating: Corpus Use for Learning to Translate and Learning Corpus Use to Translate*, Benjamins translation library, chapter 5, pages 75–107. John Benjamins Publishing Company.

Hernani Costa, Hugo Gonçalo Oliveira, and Paulo Gomes. 2010. The Impact of Distributional Metrics in the Quality of Relational Triples. In *19th European Conf. on Artificial Intelligence, Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, ECAI'10, pages 23–29, Lisbon, Portugal, August.

Hernani Costa, Hugo Gonçalo Oliveira, and Paulo Gomes. 2011. Using the Web to Validate Lexico-Semantic Relations. In *15th Portuguese Conf. on Artificial Intelligence*, volume 7026 of *EPIA'11*, pages 597–609, Lisbon, Portugal, October. Springer.

Hernani Costa, Gloria Corpas Pastor, and Miriam Seghiri. 2014. iCompileCorpora: A Web-based Application to Semi-automatically Compile Multilingual Comparable Corpora. In *Translating and the Computer 36 - AsLing*, London, UK, November.

Hernani Costa, Hanna Béchara, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. 2015a. MiniExperts: An SVM approach for Measuring Semantic Textual Similarity. In *9th Int. Workshop on Semantic Evaluation*, SemEval'15, pages 96–101, Denver, Colorado, June. ACL.

Hernani Costa, Gloria Corpas Pastor, Ruslan Mitkov, and Miriam Seghiri. 2015b. $(In\ press)$ Towards a Web-based Tool to Semi-automatically Compile, Manage and Explore Comparable and Parallel Corpora. In *7th Int. Conf. of the Iberian Association of Translation and Interpreting Studies*, AIETI, Malaga, Spain.

Hernani Costa, Gloria Corpas Pastor, Miriam Seghiri, and Ruslan Mitkov. 2015c. iCorpora: Compiling, Managing and Exploring Multilingual Data. In *7th Int. Conf. of the Iberian Association of Translation and Interpreting Studies*, AIETI, pages 74–76, Malaga, Spain, January.

Hernani Costa. 2010. Automatic Extraction and Validation of Lexical Ontologies from text. Master's thesis, University of Coimbra, Faculty of Sciences and Technology, Department of Informatics Engineering, Coimbra, Portugal, September.

Zelig Harris. 1970. Distributional Structure. In *Papers in Structural and Transformational Linguistics*, pages 775–794. D. Reidel Publishing Company, Dordrecht, Holland.

Oktay Ibrahimov, Ishwar Sethi, and Nevenka Dimitrova. 2002. The Performance Analysis of a Chi-square Similarity Measure for Topic Related Clustering of Noisy Transcripts. In *16th Int. Conf. on Pattern Recognition*, volume 4, pages 285–288. IEEE Computer Society.

Adam Kilgarriff. 2001. Comparing Corpora. *Int. Journal of Corpus Linguistics*, 6(1):97–133.

Paul Rayson, Geoffrey Leech, and Mary Hodges. 1997. Social Differentiation in the Use of English Vocabulary: Some Analyses of the Conversational Component of the British National Corpus. *Int. Journal of Corpus Linguistics*, 2(1):133–152.

Gerard Salton and Christopher Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Helmut Schmid. 1995. Improvements In Part-of-Speech Tagging With an Application To German. In *ACL SIGDAT-Workshop*, pages 47–50, Dublin, Ireland.

Amit Singhal. 2001. Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–42.

Use of Paraphrasing to Improve Matching and Retrieval in the TM

Rohit Gupta
RGCL, Research Institute of Information and Language Processing
University of Wolverhampton, Wolverhampton, UK
r.gupta@wlv.ac.uk

Abstract

Current Translation Memory (TM) systems work at the surface level and lack semantic knowledge while matching. Most of the TMs use simple edit-distance calculated on surface form or some variation of it (stem, lemma), which does not take into consideration any semantic aspects in matching. This paper presents a novel and efficient approach to incorporating semantic information in the form of paraphrasing with edit-distance. The approach is based on greedy approximation and dynamic programming. We have obtained significant improvements in recall as well as precision i.e. retrieving more segments with better quality. We have also carried out extensive human evaluation. We have measured post-editing time, keystrokes, two subjective evaluations, HTER, HMETEOR, BLEU and METEOR to substantiate our research. Our results show that paraphrasing improves TM matching and retrieval, resulting in translation performance increases when translators use paraphrase enhanced TMs.

1 Introduction

One of the core features of a TM system is the retrieval of previously translated similar segments for post-editing in order to avoid translation from scratch when an exact match is not available. However, this retrieval process is still limited to edit-distance based measures operating on surface form (or sometimes stem) matching. Most of the commercial systems use edit distance (Levenshtein, 1966) or some variation of it, e.g. the open-source TM OmegaT[1] uses word-based edit distance with some extra preprocessing. Although these measures provide a strong baseline, they are not sufficient to capture semantic similarity between the segments as judged by humans. This also results in uneven post-editing time by translators for the same fuzzy match scored segments and non-retrieval of semantically similar segments.

This paper presents a novel approach to improve matching and retrieval in TM using paraphrasing. The approach classifies paraphrases into different types for efficient implementation based on the matching of the words between the source and corresponding paraphrase. Using this approach, the fuzzy match score between segments can be calculated in polynomial time despite the inclusion of paraphrases. The method uses dynamic programming along with greedy approximation. The method calculates fuzzy match score as if the appropriate paraphrases are applied. For example, if the translation memory used has a segment "What is the actual aim of this practice ?" and the paraphrase database has paraphrases "the actual" $\Rightarrow$ "the real" and "aim of this" $\Rightarrow$"goal of this", for the input sentence "What is the real goal of this mission ?", the approach will give a 89.89% fuzzy match score (only one word, "practice", needs substitution with "mission") rather than 66.66% using simple word-based edit distance.

2 Related Work

Several researchers have used semantic or syntactic information in TMs, but the approaches were inefficient for a large TM . Their evaluations were shallow and most of the time limited to subjective evaluation carried out by the authors. This makes it hard to judge how much a semantically informed TM matching system can benefit a translator.

[1] http://www.omegat.org

Existing research (Planas and Furuse, 1999; Hodász and Pohl, 2005; Pekar and Mitkov, 2007; Mitkov, 2008) pointed out the need for similarity calculations in TMs beyond surface form comparisons. Both Planas and Furuse (1999) and Hodasz and Pohl (2005) proposed to use lemma and parts of speech along with surface form comparison. Hodasz and Pohl (2005) also extend the matching process to a sentence skeleton where noun phrases are either tagged by a translator or by a heuristic NP aligner developed for English-Hungarian translation. Planas and Furuse (1999) tested a prototype model on 50 sentences from the software domain and 75 sentences from a journal with TM sizes of 7,192 and 31,526 segments respectively. A fuzzy match retrieved was considered usable if less than half of the words required editing to match the input sentence. The authors concluded that the approach gives more usable results compared to Trados Workbench used as a baseline. Hodasz and Pohl (2005) claimed that their approach stores simplified patterns and hence makes it more probable to find a match in the TM. Pekar and Mitkov (2007) presented an approach based on syntactic transformation rules. On evaluation of the prototype model using a query sentence, the authors found that the syntactic rules help in retrieving better segments.

Recently, work by Utiyama et al. (2011) presented approach which uses paraphrasing in TM matching and retrieval. Utiyama et al. (2011) proposed an approach using a finite state transducer. They evaluate the approach with one translator and find that paraphrasing is useful for TM both in terms of precision and recall of the retrieval process. However, their approach limits TM matching to exact matches only. Simard and Fujita (2012) used different MT evaluation metrics for similarity calculation as well as for testing the quality of retrieval. For most of the metrics, the authors find that the metric which is used in evaluation gives a better score to itself (e.g. BLEU gives highest score to matches retrieved using BLEU as similarity measure).

3 Our Approach

In this section, we present the approach to include paraphrasing in the TM matching and retrieval process. A trivial approach to implementing paraphrasing along with edit-distance is to generate all the paraphrases based on the paraphrases available and store these additional segments in the TM. This approach is highly inefficient both in terms of time and space. For example, for a TM segment which has four different phrases where each phrase can be paraphrased in five more possible ways, we get 1295 (6^4 -1) additional segments (still not considering that these phrases may contain paraphrases as well) to store in the TM, which is inefficient even for small TMs. To handle this problem, we classify paraphrases and process each class of paraphrases in a different manner.

3.1 Classification of Paraphrases

We have used the lexical and phrasal paraphrases from the PPDB corpus (Ganitkevitch et al., 2013) of L size. We have classified paraphrases obtained from PPDB 1.0 into four types for our implementation on the basis of the number of words in the source and target phrases. These four categories are as follows:

1. Paraphrases having one word on both the source and target sides, e.g. “period” ⇒“duration”

2. Paraphrases having multiple words on both sides but differing in one word only, e.g. “in the period” ⇒ “during the period”

3. Paraphrases having multiple words as well as the same number of words on both sides, e.g. “laid down in article” ⇒ “set forth in article”

4. Paraphrases in which the number of words on the source and target sides differ, e.g. “a reasonable period of time to” ⇒ “a reasonable period to”

In our classification, Type 1 are one-word paraphrases and Type 2 can be reduced to one-word paraphrases after considering the context when storing in the TM. For Type 1 and Type 2, we get the same accuracy as the trivial method in polynomial time complexity (see Section 3.2 for details). Paraphrases of Type 3 and Type 4 require additional attention because they still remain multiword paraphrases after reduction and greedy approximation is needed to implement them in polynomial time.

Type 1 paraphrases appear to be simple synonyms (e.g. WordNet) but they are better than simple synonyms. They happen to be one word because additional context was not required for them. In addition, in typical TM settings we are interested in more than 70% fuzzy match, which up to a certain extent makes sure that apart from paraphrases some other context words appear when a match is found in the TM. Furthermore, we also use filtering steps (explained in Section 3.4) to restrict the amount of paraphrasing allowed per segment.

3.2 Matching Steps

There are two options for incorporating paraphrasing in a typical TM matching pipeline: paraphrase the input or paraphrase the TM. For our approach we have chosen to paraphrase the TM. There are many reasons for this. First, once a system is set up, the user can get the retrieved matches in real time; second, TMs can be stored in company servers and all processing can be done offline; third, the TM system need not be installed on the user computer and can be provided as a service.

Paraphrasing the input has its own advantage. In general, input files are much shorter than TM. Therefore, paraphrasing the input instead of TM can save space.

For our implementation we used basic edit-distance implementation of OmegaT, which uses word-based edit-distance with cost 1 for insertion, deletion and substitution. We have employed this edit-distance as a baseline and adapted this to incorporate paraphrasing.

Our approach can be briefly described as the following steps:

1. Read the Translation Memories available
2. Collect all the paraphrases from the paraphrase database and classify them according to the classes presented in Section 3.1
3. Store all the paraphrases for each segment in the TM in their reduced forms according to the process presented in Section 3.3
4. Read the file that needs to be translated
5. For each segment in the input file get the potential segments for paraphrasing in the TM according to the filtering steps of Section 3.4 and search for the most similar segment based on approach described in Section 3.5 and retrieve the most similar segment if above a predefined threshold

3.3 Storing Paraphrases

The paraphrases are stored in the TM in their reduced forms as after capturing paraphrases for a particular segment we have already considered the context, and there is no need for it to be considered again while calculating edit-distance. We store only the uncommon substring instead of the whole paraphrase. This reduced paraphrase is stored with the source word where the uncommon substring starts. Suppose we have paraphrases as given in Table 1. The paraphrases are stored in a reduced form as given in Table 2.

the period laid down in	the period referred to in
laid down in article	provided for by article
period	time
period	duration

Table 1: Example: Paraphrases

3.4 Filtering

Before processing begins, for each input segment certain filtering steps are applied in order to speed up the process. The purpose of this preprocessing is to filter out unnecessary candidates for participating in the paraphrasing process. Because we are generally interested in candidates above a certain threshold it is obvious to filter out candidates below a certain threshold. Our filtering steps for getting potential candidates for paraphrasing are as follows:

the	period			laid	down in article	4(3)	of	decision	468
the	period / duration / time	ls	tp	laid	down in article	4(3)	of	decision	468
		2		referred to					
		3		provided for by					

Table 2: Representing paraphrases in the TM

- We first filter out the segments based on length because if segments differ considerably in length, the edit-distance will also differ. In our case, the threshold for length was 39%. So, the TM segments which are shorter than 39% of the input are filtered.
- Next, we filter out the segments based on baseline edit-distance similarity. The TM segments which have a similarity below a certain threshold will be removed. In our case, the threshold was 39%.
- Next, after filtering the candidates with the above two steps we sort the remaining segments in decreasing order of similarity and pick the top 100 segments.
- Finally segments within a certain range of similarity with the most similar segment were selected for paraphrasing. In our case, the range is 35%. This means that if the most similar segment has 95% similarity, segments with a similarity below 60% will be discarded[2].

For matching, similarity is calculated with the potential segments for paraphrasing extracted as per Section 3.4. Type 1 and Type 2 paraphrases after reduction (see Section 2) are single-word paraphrases and Type 3 and Type 4 paraphrases have multiple words. For Type 1 and Type 2 the edit-distance procedure can be optimised globally as this is a simple case of matching one of these "paraphrases" when calculating the cost of substitution.

	j	0	1	2				3	4				5
i		#	the	period duration time	laid	down	in	referred	to	provided	for	by	in
0	#	0	1	2	3	4	5	3	4	3	4	5	5
1	the	1	0	1	2	3	4	2	3	2	3	4	4
2	period	2	1	0	1	2	3	1	2	1	2	3	3
3	referred	3	2	1	1	2	3	0	1	1	2	3	2
4	to	4	3	2	2	**2**	3	1	**0**	2	2	3	1
5	in	5	4	3	3	3	**2**	2	1	3	3	**3**	0

Table 3: Edit-Distance Calculation

3.5 Edit-Distance calculation

Table 3 provides an example of edit-distance calculation with paraphrasing. The second column represents input segment and the second row represents the TM segment along with the paraphrases collected as given in Table 2. In Table 3, if a word from the input segment matches any of the words "period", "time" or "duration", the cost of substitution will be 0. For paraphrases of Types 3 and 4 the algorithm takes the decision locally at the point where all paraphrases finish. As we can see in Table 3, starting from the third token of the TM, "laid", three separate edit-distances are calculated, two for the two paraphrases "referred to" and "provided for by" and one for the corresponding longest source phrase

[2]these thresholds were determined empirically

"laid down in", and the paraphrase "referred to" is selected as it gives a minimum edit-distance of 0. The last column of Table 3 ($j = 5$) shows the edit-distance calculation of the next token "in" after selecting "referred to". The more detailed algorithm is given in Gupta and Orăsan (2014)

3.6 Computational Considerations

The time complexity of the basic edit-distance procedure is $O(mn)$ where m and n are lengths of source and target segments, respectively. After employing paraphrasing of Type 1 and Type 2 the complexity of calculating the substitution cost increases from $O(1)$ to $O(log(p))$ (as searching p words takes $O(log(p))$ time) where p is the number of paraphrases of Type 1 and Type 2 per token of TM source segment, which increases the edit-distance complexity to $O(mnlog(p))$. Employing paraphrasing of Type 3 and Type 4 further increases the edit-distance complexity to $O(lmn(log(p) + q))$, where q is the number of Type 3 and Type 4 paraphrases stored per token and l is the average length of paraphrase. Assuming the source and target segment are of the same length n and each token of the segment stores paraphrases of length l, the complexity will be $O((q + log(p))n^2l)$. By limiting the number of paraphrases stored per token of the TM segment we can replace $(q + log(p))$ by a constant c. In this case complexity will be $c \times O(n^2l)$. However, in practice it will take less time as not all tokens in the TM segment will have p and q paraphrases and the paraphrases are also stored in the reduced form.

4 Evaluation

In TM, the performance of retrieval can be measured by counting the number of segments or words retrieved. However, NLP techniques are not 100% accurate and most of the time, there is a tradeoff between the precision and recall of this retrieval process. This is also one of the reasons that TM developers shy away from using semantic matching. One cannot measure the gain unless retrieval benefits the translator.

When we use paraphrasing in the matching and retrieval process, the fuzzy match score of a paraphrased segment is increased, which results in the retrieval of more segments at a particular threshold. This increment in retrieval can be classified in two types: without changing the top rank; and by changing the top rank. For example, for a particular input segment, we have two segments: A and B in the TM. Using simple edit-distance, A has a 65% and B has a 60% fuzzy score; the fuzzy score of A is better than that of B. As a result of using paraphrasing we notice two types of score changes:

1. the score of A is still better than or equal to that of B, for example, A has 85% and B has 70% fuzzy score;

2. the score of A is less than that of B, for example, A has 75% and B has 80% fuzzy score.

In the first case, paraphrasing does not supersede the existing model and just facilitates it by improving the fuzzy score so that the top segment ranked using edit distance gets retrieved. However, in the second case, paraphrasing changes the ranking and now the top-ranked segment is different. In this case, the paraphrasing model supersedes the existing simple edit distance model. This second case also gives a different reference with which to compare. We take the top segment retrieved using simple edit distance as a reference against the top segment retrieved using paraphrasing and compare to see which is better for a human translator to work with.

4.1 Post-editing Time (PET) and Keystrokes (KS)

In this evaluation, the translators were presented with fuzzy matches and the task was to post-edit the segment in order to obtain a correct translation. The translators were presented with an input English segment, the German segment retrieved from the TM for post-editing and the English segment used for matching in the TM.

In this task, we recorded post-editing time (PET) and keystrokes (KS). The post-editing time taken for the whole file is calculated by summing up the time taken on each segment. Only one segment is visible on screen. The segment is only visible after clicking and the time is recorded from when the segment becomes visible until the translator finishes post-editing and goes to the next screen. The next screen is a

blank screen so that the translator can have a rest after post-editing a segment. The translators were aware that the time is being recorded. Each translator post-edited half of the segments retrieved using simple edit distance (ED) and half of the segments retrieved using paraphrasing (PP). The ED and PP matches were presented one after the other (ED at odd positions and PP at even positions or vice versa). However, the same translator did not post-edit the match retrieved using PP and ED for the same segment; instead five different translators post-edited the segment retrieved using PP and another five different translators post-edited the match retrieved using ED.

Post-editing time (PET) for each segment is the mean of the normalised time (N) taken by all translators on this segment. Normalisation is applied to account for both slow and fast translators.

$$\text{PET}_j = \frac{\sum_{i=1}^{n} N_{ij}}{n} \tag{1}$$

$$N_{ij} = T_{ij} \times \frac{\text{Avg time on this file by all translators}}{\sum_{j=1}^{m} T_{ij}} \tag{2}$$

In the equations 1 and 2 above, PET_j is the post-editing time for each segment j, n is the number of translators, N_{ij} is the normalised time of translator i on segment j, m is the number of segments in the file, and T_{ij} is the actual time taken by a translator i on a segment j.

Along with the post-editing time, we also recorded all printable keystrokes, whitespace and erase keys pressed. For our analysis, we considered average keystrokes pressed by all translators for each segment.

4.2 Subjective Evaluation with Two Options (SE2)

In this evaluation, we carried out subjective evaluation with two options (SE2). We presented fuzzy matches retrieved using both paraphrasing (PP) and simple edit distance (ED) to the translators. The translators were unaware of the details (ED or PP) of how the fuzzy matches were obtained. To neutralise any bias, half of the ED matches were tagged as A and the other half as B, with the same applied to PP matches. The translator has to choose between two options: whether A is better; or B is better. 17 translators participated in this experiment. Finally, the decision of whether 'ED is better' or 'PP is better' is made on the basis of how many translators choose one over the other.

4.3 Subjective Evaluation with Three Options (SE3)

This evaluation is similar to Evaluation SE2 except that we provided one more option to translators. Translators can choose among three options: whether A is better; B is better; or both are equal. 7 translators participated in this experiment.

5 Corpus, Tool and Translators expertise

As a TM and test data, we have used English-German pairs of the Europarl V7.0 (Koehn, 2005) corpus with English as the source language and German as the target language.[3] From this corpus we have filtered out segments of fewer than seven words and greater than 40 words to create the TM and test datasets. Tokenization of the English data was done using the Berkeley Tokenizer (Petrov et al., 2006).

	TM	Test Set
Segments	1565194	9981
Source words	37824634	240916
Target words	36267909	230620

Table 4: Corpus Statistics

In these experiments, we have not paraphrased any capitalised words (but we lowercase them for both baseline and paraphrasing similarities calculation). This is to avoid paraphrasing any named entities. Table 4 shows our corpus statistics.

[3]The results on English-French pairs from DGT-TM corpus are given in Gupta and Orăsan (2014).

The translators involved in our experiments were third-year bachelor or masters translation students who were native speakers of German with English language level C1, in the age group of 21 to 40 years with a majority of female students. Our translators were not expert in any specific technical or legal field. For this reason we did not use such a corpus. In this way we avoid any bias from unfamiliarity or familiarity with domain specific terms.

5.1 Familiarisation with the Tool

We used the PET tool (Aziz et al., 2012) for all our human experiments. However, settings were changed depending on the experiment. To familiarise translators with the PET tool we carried out a pilot experiment before the actual experiment with the Europarl corpus. This experiment was done on a corpus (Vela et al., 2007) different from Europarl. 18 segments are used in this experiment. While the findings are not included in this paper, they informed the design of our main experiments.

5.2 Results and Analysis

The retrieval results are given in Table 5 and Table 6. Table 5 presents the results using a cutoff threshold. whereas Table 6 shows the interval wise results. We have chosen the threshold intervals so as to select the segments from each range for the human evaluations.

Tables 5 and 6 show similarity thresholds for TM (TH), the total number of segments retrieved using the baseline approach (EditRetrieved), the additional number of segments retrieved using the paraphrasing approach (+ParaRetrieved), the percentage improvement in retrieval obtained over the baseline (%Improve), and the number of segments that changed their ranking and rose to the top because of paraphrasing (RankCh). BLEU-ParaRankCh and METEOR-ParaRankCh represents the BLEU score (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014) score over translations retrieved by our approach for segments which changed their ranking and come up to the top because of paraphrasing and BLEU-EditRankCh and METEOR-EditRankCh represent the BLEU score and METEOR score on corresponding translations retrieved by baseline approach. NumPara and NumParaRankCh in Table 6 represent the number of unique paraphrases used to retrieve +ParaRetrieved and RankCh, respectively.

TH	**100**	**95**	**90**	**85**	**80**	**75**	**70**
EditRetrieved	117	127	163	215	257	337	440
+ParaRetrieved	16	16	22	33	49	79	102
%Improve	13.68	12.6	13.5	15.35	19.07	23.44	23.18
RankCh	9	10	16	25	36	65	97
BLEU-EditRankCh	31.88	32.37	27.70	21.71	19.32	14.98	12.25
BLEU-ParaRankCh	**52.00**	**47.92**	**43.90**	**31.76**	**25.24**	**19.75**	**15.28**
METEOR-EditRankCh	45.48	46.48	45.59	39.24	37.32	34.02	31.10
METEOR-ParaRankCh	**68.08**	**67.03**	**61.09**	**50.07**	**44.16**	**38.35**	**33.19**

Table 5: Results on Europarl dataset: Automatic Evaluation, using all four types of paraphrases

TH	**100**	**[85, 100)**	**[70, 85)**	**[55, 70)**
EditRetrieved	117	98	225	703
+ParaRetrieved	16	30	98	311
%Improve	13.67	30.61	43.55	44.23
RankCh	9	14	55	202
BLEU-EditRankCh	31.88	13.18	6.85	5.32
BLEU-ParaRankCh	**52.00**	**17.10**	**8.37**	**5.60**
METEOR-EditRankCh	45.48	34.37	25.76	20.05
METEOR-ParaRankCh	**68.08**	**40.00**	**25.82**	**21.69**
NumPara	24	49	169	535
NumParaRankCh	14	24	92	356

Table 6: Results of Retrieval

We can see in Table 5 and 6 that we get improvements on each threshold level and intervals. Table 6 shows that when using paraphrasing we obtain around 13.67% improvement in retrieval for exact matches and more than 30% and 43% improvement in the intervals [85, 100) and [70, 85), respectively.

This clearly shows that paraphrasing significantly improves the retrieval results. We have also observed that there are different paraphrases used to bring this improvement. As given in Table 6, in the interval [70, 85), 169 different paraphrases are used to retrieve 98 more segments.

The sets' distribution for human evaluation is given in Table 7. The sets contain randomly selected segments from the additionally retrieved segments using paraphrasing who changed their top ranking.

TH	100	[85, 100)	[70, 85)	Total
Set1	2	6	6	14
Set2	5	4	7	16
Total	7	10	13	30

Table 7: Test Sets for Experiments PET, KS, SE2 and SE3

Results for human evaluations (PET, KS, SE2 and SE3) on both sets (Set1 and Set2) are given in Table 8. Here 'Seg #' represents the segment number, 'ED' represents the match retrieved using simple edit distance and 'PP' represents the match retrieved after incorporating paraphrasing. 'EDB', 'PPB' and 'BEQ' in Subjective Evaluations represent the number of translators prefer the 'ED is better', 'PP is better' and 'Both are equal' options respectively.

	Post-editing				Subjective Evaluations				
	PET		KS		SE2 (2 Options)		SE3 (3 options)		
Seg #	ED	PP	ED	PP	EDB	PPB	EDB	PPB	BEQ
1	42.98	41.30↑	42.4	**0.4**↑	**1**	**16**↑	0	7↑	0
2!+	13.72	10.65↑	2.8	2.4↑	10	7↓	2	2	3
3*!	13.88	12.62↑	2.0	3.6↓	12	5↓	4	1↓	2
4	37.97	**17.64**↑	26.2	**6.2**↑	**1**	**16**↑	0	6↑	1
5!+	21.52	17.69↑	22.4	13.2↑	**13**	**4**↓	2	3↑	2
6!+	41.14	42.74↓	13.2	34.4↓	**4**	**13**↑	2	0	5
7!+	33.69	31.59↑	34.0	33.4↑	10	7↓	1	0	6
8	47.14	**23.41**↑	61.6	**6.4**↑	**0**	**17**↑	0	7↑	0
9	22.89	**14.20**↑	37.2	**2.2**↑	**0**	**17**↑	0	6↑	1
10	46.89	38.20↑	77.6	65.6↑	**1**	**16**↑	0	1	6
11	58.25	53.65↑	82.8	58.8↑	**0**	**17**↑	0	3	4
12!+	34.04	45.03↓	36.8	39.6↓	**2**	**15**↑	0	6↑	1
13	30.34	**21.12**↑	54.8	39.2↑	7	10↑	1	1	5
14!+	75.50	96.54↓	38.8	50.8↓	5	12↑	0	3	4
Set1-subtotal	520.02	466.44	532.60	356.20	66	172	12	46	40
15	24.14	**9.18**↑	24.0	**0.0**↑	5	12↑	1	5↑	1
16*+	28.30	29.20↓	23.4	15.4↑	11	6↓	2	2	3
17*!	65.64	53.49↑	6.2	22.4↓	10	7↓	2	3↑	2
18	41.91	**20.98**↑	28.0	**2.0**↑	**1**	**16**↑	0	6↑	1
19	29.81	19.71↑	23.8	**6.8**↑	7	10↑	2	3↑	2
20	41.25	**15.42**↑	39.0	**3.8**↑	**0**	**17**↑	1	5↑	1
21*!	**42.04**	65.44↓	39.4	36.0↑	7	10↑	1	2	4
22	29.28	35.87↓	17.0	33.4↓	12	5↓	5	0↓	2
23	**32.64**	49.49↓	**11.4**	50.8↓	11	6↓	2	2	3
24!+	59.35	54.54↑	79.6	79.2↑	**17**	**0**↓	5	0↓	2
25	62.51	61.30↑	71.0	54.0↑	**2**	**15**↑	0	3	4
26*!	36.82	41.06↓	55.0	**23.4**↑	**1**	**16**↑	0	6↑	1
27!+	**27.21**	44.02↓	**24.4**	48.8↓	**4**	**13**↑	1	5↑	1
28	40.99	**33.08**↑	39.6	**24.6**↑	5	12↑	3	4↑	0
29	52.01	**31.55**↑	50.6	**23.4**↑	**2**	**15**↑	0	6↑	1
30*!	43.76	38.76↑	38.2	44.6↓	**15**	**2**↓	1	1	5
Set2-subtotal	657.75	603.17	570.6	468.59	110	162	26	53	33
Total	1177.77	1069.61	1103.2	824.79	176	334	38	99	73

Table 8: Results of Human Evaluation on Set1 (1-14) and Set2 (15-30)

5.3 Results: Post-editing Time (PET) and Keystrokes (KS)

As we can see in Table 8, improvements were obtained for both sets. ↑ demonstrates cases in which PP performed better than ED and ↓ shows where ED performed better than PP. Entries in bold for PET, KS and SE2 indicate where the results are statistically significant [4].

[4] $p<0.05$, one tailed Welch's t-test for PET and KS, χ^2 test for SE2. Because of the small sample size for SE3, no significance test was performed on individual segment basis.

For Set1, translators made 356.20 keystrokes and 532.60 keystrokes when editing PP and ED matches, respectively. Translators took 466.44 seconds for PP as opposed to 520.02 seconds for ED matches. This means that by using PP matches, translators edit 33.12% less (49.52% more using ED), which saves 10.3% time .

For Set2, translators made 468.59 keystrokes and 570.6 keystrokes when editing PP and ED matches respectively. Translators took 603.17 seconds for PP as opposed to 657.75 seconds for ED matches. This means that by using PP matches, translators edit 17.87% less (21.76% more using ED), which saves 8.29% time.

In total, combining both the sets, translators made 824.79 keystrokes and 1103.2 keystrokes when editing PP and ED matches, respectively. Translators took 1069.61 seconds for PP as opposed to 1177.77 seconds for ED matches. Therefore, by using PP matches, translators edit 25.23% less, which saves time by 9.18%. In other words, ED matches require 33.75% more keystrokes and 10.11% more time. We observe that the percentage improvement obtained by keystroke analysis is smaller compared to the improvement obtained by post-editing time. One of the reasons for this is that the translator spends a fair amount of time reading a segment before starting editing.

5.4 Results: Using post-edited references

We also calculated the human-targeted translation error rate (HTER) (Snover et al., 2006) and human-targeted METEOR (HMETEOR) (Denkowski and Lavie, 2014). HTER and HMETEOR was calculated between ED and PP matches presented for post-editing and references generated by editing the corresponding ED and PP match. Table 9 lists HTER5 and HMETEOR5, which use five corresponding ED or PP references only and HTER10 and HMETEOR10, which use all ten references generated using ED and PP.

Table 9 shows improvements in both the HTER5 and HMETEOR5 scores. For Set-1, HMETEOR5 improved from 59.82 to 81.44 and HTER5 improved from 39.72 to 17.63[5]. For Set-2, HMETEOR5 improved from 69.81 to 80.60 and HTER5 improved from 27.81 to 18.71. We also observe that while ED scores of Set1 and Set2 differ substantially (59.82 vs 69.81 and 39.72 vs 27.81), PP scores are nearly the same (81.44 vs 80.60 and 17.63 vs 18.71). This suggests that paraphrasing not only brings improvement but may also improve consistency.

	Set-1		Set-2	
	ED	**PP**	**ED**	**PP**
HMETEOR5	59.82	81.44	69.81	80.60
HTER5	39.72	17.63	27.81	18.71
HMETEOR10	59.82	81.44	69.81	80.61
HTER10	36.93	18.46	27.26	18.40

Table 9: Results using human targeted references

5.5 Results: Subjective evaluations

The subjective evaluations also show significant improvements.

In subjective evaluation with two options (SE2) as given in Table 8, from a total of 510 (30×17) replies for 30 segments from both sets by 17 translators, 334 replies tagged 'PP is better' and 176 replies tagged 'ED is better' [6].

In subjective evaluation with three options (SE3), from a total of 210 (30×7) replies for 30 segments from both sets by 7 translators, 99 replies tagged 'PP is better', 73 replies tagged 'both are equal' and 38 replies tagged 'ED is better' [7].

[5]For HMETEOR, higher is better and for HTER lower is better

[6]statistically significant, χ^2 test, $p < 0.001$

[7]statistically significant, χ^2 test, $p < 0.001$

5.6 Results: Segment wise analysis

A segment wise analysis of 30 segments from both sets shows that 21 segments extracted using PP were found to be better according to PET evaluation and 20 segments using PP were found to be better according to KS evaluation. In subjective evaluations, 20 segments extracted using PP were found to be better according to SE2 evaluation whereas 27 segments extracted using PP were found to be better or equally good according to SE3 evaluation (15 segments were found to be better and 12 segments were found to be equally good).

We have also observed that not all evaluations correlate with each other on segment-by-segment basis. '!, '+ and '* next to each segment number in Table 8 indicate conflicting evaluations: '!' denotes that PET and SE2 contradict each other, '+' denotes that KS and SE2 contradict each other and '*' denotes that PET and KS contradict each other. In twelve segments where KS evaluation or PET evaluation show PP as statistically significant better, except for two cases all the evaluations also shows them better.[8] For Seg #13 SE3 shows 'Both are equal' and for Seg #26, PET is better for ED, however for these two sentences also all the other evaluations show PP as better.

In three segments (Seg #'s 21, 23, 27) KS evaluation or PET evaluation show ED as statistically significant better, but none of the segments are tagged better by all the evaluations. In Seg #21 all the evaluations with the exception of PET show PP as better. In Seg #23, SE3 shows 'both are equal'. Seg #23 is given as follows:

Input: The next item is the Commission declaration on Belarus .

ED: The next item is the Commission Statement on AIDS .//Als nächster Punkt folgt die Erklärung der Kommission zu AIDS.

PP: The next item is the Commission statement on Haiti .//Nach der Tagesordnung folgt die Erklärung der Kommission zu Haiti.

In Seg #23, apart from "AIDS" and "Haiti" the source side does not differ but the German side differs. The reason for PP match retrieval was that "statement on" in lower case was paraphrased as "declaration on" while in the other segment "Statement" was capitalised and hence was not paraphrased. If we look at the German side of both ED and PP, "Nach der Tagesordnung" requires a broader context to accept it as a translation of "The next item" whereas "Als nächster Punkt" does not require much context.

In Seg #27, we observe contradictions between post-editing evaluations and subjective evaluations. Seg #27 is given below (EDPE and PPPE are post-edited translations of ED and PP match respectively):

Input: That would be an incredibly important signal for the whole region .

ED: That could be an important signal for the future .//Dies könnte ein wichtiges Signal für die Zukunft sein.

PP: That really would be extremely important for the whole region .//Und das wäre wirklich für die ganze Region extrem wichtig.

EDPE: Dies könnte ein unglaublich wichtiges Signal für die gesamte Region sein.

PPPE: Das wäre ein unglaublich wichtiges Signal für die ganze Region.

In subjective evaluations, translators tagged PP as better than ED. But, post-editing suggests that it takes more time and keystrokes to post-edit the PP compare to ED.

There is one segment, Seg #22, on which all the evaluations show that ED is better. Seg #22 is given below:

Input: I would just like to comment on one point.

[8] In this section all evaluations refer to all four evaluations viz PET, KS, SE2 and SE3.

ED: I would just like to emphasise one point.//Ich möchte nur eine Sache betonen.

PP: I would just like to concentrate on one issue.//Ich möchte mich nur auf einen Punkt konzentrieren.

In segment 22, the ED match is clearly closer to the input than the PP match. Paraphrasing "on one point" as "on one issue" does not improve the result. Also, "konzentrieren" being a long word takes more time and keystrokes in post-editing.

6 Conclusion

We conclude that paraphrasing significantly improves retrieval. We observe more than 30% and 43% improvement for the threshold intervals [85, 100) and [70, 85), respectively and more than 23% improvement over 70 or 75 cutoff threshold. The quality of the retrieved segment is also significantly better, which is evident from all our human translation evaluations. On average on both sets used for evaluation, compared to paraphrasing simple edit distance takes 33.75% more keystrokes and 10.11% more time when evaluating the segments who changed their top rank and come up in the threshold intervals because of paraphrasing.

Acknowledgements

Rohit Gupta is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471.

References

Wilker Aziz, S Castilho, and Lucia Specia. 2012. PET: a Tool for Post-editing and Assessing Machine Translation. *Language Resources and Evaluation*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Juri Ganitkevitch, Van Durme Benjamin, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia.

Rohit Gupta and Constantin Orăsan. 2014. Incorporating Paraphrasing in Translation Memory Matching and Retrieval. In *Proceedings of the European Association of Machine Translation (EAMT-2014)*.

Gábor Hodász and Gábor Pohl. 2005. MetaMorpho TM: a linguistically enriched translation memory. In *In International Workshop, Modern Approaches in Translation Technologies*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Ruslan Mitkov. 2008. Improving Third Generation Translation Memory systems through identification of rhetorical predicates. In *Proceedings of LangTech2008*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Viktor Pekar and Ruslan Mitkov. 2007. New Generation Translation Memory: Content-Sensivite Matching. In *Proceedings of the 40th Anniversary Congress of the Swiss Association of Translators, Terminologists and Interpreters*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the COLING/ACL*, pages 433–440.

Emmanuel Planas and Osamu Furuse. 1999. Formalizing Translation Memories. In *Proceedings of the 7th Machine Translation Summit*, pages 331–339.

Michel Simard and Atsushi Fujita. 2012. A Poor Man s Translation Memory Using Machine Translation Evaluation Metrics . In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. 2011. Searching Translation Memories for Paraphrases. In *Machine Translation Summit XIII*, pages 325–331.

Mihaela Vela, Stella Neumann, and Silvia Hansen-Schirra. 2007. Querying multi-layer annotation and alignment in translation corpora. In *Proceedings of the Corpus Linguistics Conference CL*.

EXPERT Innovations in Terminology Extraction and Ontology Induction

Liling Tan
Universität des Saarlandes
Campus A2.2, Germany
liling.tan@uni-saarland.de

Abstract

This paper reports the innovations made to terminology extraction and ontology induction under the EXPERT project. We proposed (i) a novel approach to terminology extraction using pre-trained language model and a pointwise mutual information based criteria and (ii) a new unsupervised ontological induction approach using vector-space word embeddings of a non-content word vector. Our preliminary experiments have shown promising results of the new approaches and we discuss the ongoing work during the last phase of the EXPERT endeavours in applying the new term extraction and ontology induction methods to improve translation quality.

1 Introduction

This paper reports the innovations made to terminology extraction and ontology induction under the EXPERT project. We present (i) a novel LM-PMI term extraction algorithm using a pre-trained language model (LM) which produces a list of term candidates ranked by their cumulative Pointwise Mutual Information (PMI) and (ii) an unsupervised ontology induction system using neural network word embeddings of a non-content word vector.

The rest of the article will split into three parts, (i) describing the LM-PMI terminology extraction algorithm, (ii) the vector-space ontology induction system and (iii) concludes the paper by describing ongoing work to apply the novel innovations to improve translation quality.

2 Terminology Extraction

A **term** is the *designation of a defined concept in a special language by a linguistic expression*; a term may consist of one or more words. A **terminology** refers to the set of terms representing the system of concepts of a particular subject field (ISO 1087). The International Organization of Standardization (ISO) history of terminology traces back to Wüster (1969) seminal article on *Die vier Dimensionen der Terminologiearbeit*[1] which the ISO Technical Committee 37 (ISO/TC 37) builds upon in providing the common standards related to terminology work.

A later formulation states that a term is *any conventional symbol representing a concept defined in a subject field*; a terminology is the aggregate of terms, which represent the system of concepts of an individual subject field (Flber, 1984). The core characteristic of a term is defined as **termhood**, i.e. *the degree to which a linguistic unit is related to a domain-specific context* (Kageura and Umino, 1996). In the case of multi-token terms, additional substantiation is necessary to check its **unithood**, i.e. *the degree of strength or stability of syntagmatic combinations and collocations* (Kageura and Umino, 1996).

Single token terms can be perceived as a specialized vocabulary that is used specifically in a domain. The surface word representing the single token term is often polysemous and the usage of the term within a specialized domain may narrow down the set of possible senses or single out a disambiguated sense of the word.

For example, the term "*classifier*" can refer to (i) *a morpheme used to indicate the semantic class to which the counted item belongs* or (ii) *a pre-trained model to identify/distinguish different classes*

[1]The Four Dimensions of Terminological Work

within a dataset. The first definition is mainly used within linguistic research, the second within the machine learning domain. However, when "*classifier*" is used in computational linguistics, its usage is ambiguous. The latter definition of "*classifier*" tends to be used more often than the former.

In English, terms are more often multi-word expressions (MWE), primarily nominal phrases, made up of a head noun and its complement adjective(s), prepositional clause(s), or compounding noun(s). Commonly, a complex term can be analysed in terms of a head with one or more modifiers (Hippisley et al., 2005).

2.1 Rule-based Term Extraction

The linguistic properties of a term can be characterized by its syntactic context, previous approaches to term extraction use these linguistics properties in form of Part-Of-Speech (POS) patterns. For example, Justeson and Katz (1995) and Daille (2000) used the following POS patterns to extract nominal phrasal terms:

EN: ((Adj|(Noun)+|((Adj|Noun)*(NounPrep)?)(Adj|Noun)*)Noun$_{head}$

FR: Noun$_{head}$ (Adj|(Prep(Det)?)?Noun |V$_{inf}$)

In the case of English, the compulsory head noun is in the final position preceded by its modifiers whereas in French, it is in the first position followed by its modifiers. Tan et al. (2015c) observe that the multi-word nature of Romance languages produces more terminological phrases whereas for Germanic languages, the compounding nature of nouns derives more single token lexicalized terms. For example, an equivalent POS pattern for German would have to be replaced by a combination of POS and morphemic pattern:

DE: ((AC|NC)+|((AC|NC)*(Noun|Prep)?)(AC|NC)*)Noun$_{head}$

Similar to the (Adj|Noun) pattern in English, the German ((AC|NC) pattern is a combination of adjective/noun with occasional connective morpheme where an connective morpheme might be necessary to join the adjacent adjectives/nouns. For example, in the German compound noun "*Mausefalle*" (*Mousetrap*), the "*Falle*" (*trap*) is head noun in the final position and the word "*Maus*" (*mouse*) attaches to the head noun with the "*-e-*" connective morpheme between the nouns.

2.2 Statistical Term Extraction

The basis of all statistical properties in multi-word term extraction relies on the frequency of a token or an n-gram in a corpus. Frequency counts are combined to compute co-occurrence measures (aka. word/lexical association measures) that quantify the probabilistic occurrence of a word with its neighbouring words. Co-occurrence measures are used to estimate the propensity for words occurring together. Psycholinguistic evidences show that word association norms can be measured as a subjects responses to words when preceded by associated words (Palermo and Jenkins, 1964) and humans respond quicker in the case of highly associated words within the same domain (Church and Hanks, 1990).

Common co-occurrence measures, e.g. Dice coefficient, Mutual Information (MI), Pointwise Mutual Information (PMI), Log-Likelihood Ratio (LLR) and Phi-square (ϕ^2) rely on three types of frequency information; (i) the frequency of a word occurring in the corpus, (ii) the joint frequency of a word occurring with another word, (iii) the total number of words in the corpus (Tan et al., 2015c).

2.3 C-Value

Frantzi et al. (2000) introduced a method to use both linguistic and statistical information using the C-value. They start with a set of POS patterns and a stop word list to pre-filter possible n-grams before they calculate the n-grams termhood using the C-value metric and the concept of nested terms. Nested terms refer to those terms that appear within other longer terms and may or may not appear by themselves in the corpus, e.g. *floating point* is a nested term because it is also found in *floating point arithmetic*.

For non-nested terms, the C-value accounts for the length of the term candidate and its frequency. For nested terms, the C-value subtracts the average number of times the term is nested in other term n-grams.

Thus if floating point occurs as a nested term candidate as often or more than it does as an independent term, then it will have low C-value.

2.4 Language Model - Pointwise Mutual Information (LM-PMI) Value

The calculations for frequencies described in the statistical term extraction is based on raw counts of a word or term. Instead of using raw counts, we propose the usage of pre-trained n-gram language models.

N-gram language models have developed and applied to other NLP applications such as speech processing and machine translation. The major advantage of using a language model is the possibility of accounting for unknown words using interpolation and smoothing techniques (Chen and Goodman, 1996). By using a language model, we avoid the need to optimize n-gram counting when implementing the term extraction algorithm, especially when very fast implementations of language models already exists (Heafield, 2011).

The Pointwise Mutual Information (PMI) of any term can be calculated with a backoff trigram language model, formally we describe it as such: let LM-PMI(t) be the pointwise mutual information (PMI) score based on language model termhood probability of a term t occurring. Let i be a possible n-gram in a term such that $i \in t$ and $t = <i_1, \ldots\ i_n>$, where n is the total no. of possible n-grams, excluding unigrams, in t. Let u be any unknown word not seen in the corpus, represented by $<unk>$.

$$\text{LM-PMI}(t) = 1/\text{n} \sum_{i \in t} \begin{cases} log\ PMI(i_1, i_2) & \text{if all words in term in found in training corpus} \\ PMI_{LM}(u, i) & \text{Otherwise, when } i \text{ is an unknown word} \end{cases}$$

The brevity normalization (1/n) does not favour a longer term compared to the C-value. When an unknown word exists in the term, it uses the backoff probability with the contextual probability of an unknown word occurring in the i_1 and i_2 context.

2.5 Evaluation

We evaluate our novel approach on the food domain corpus (WikiFood) that was built for an ontology induction task at SemEval-2015. The corpus contains 869 food terms and the relevant Wikipedia articles that contain these terms. Of those terms 752 terms are multi-words and from the 752 multi-words, we extracted 42,851 sentences (1,207,677 tokens) that contains the multi-word terms. We expect only 1 correct term to be extracted per sentence.

To access the probabilities for calculating LM-PMI, we trained a 5-gram language model on the corpus and we evaluate the LM-PMI accuracy against the traditional C-value score in extracting terms from the corpus.

We extract the top five term candidates each using LM-PMI and C-value to match against the 1 correct term per sentence. We evaluate the metrics by calculating the accuracy of the top ranked term candidates for each sentence and matching them against the correct term for that sentence. Since the experimental task is structured more like an information retrieval task of candidate ranking, we use the mean reciprocal rank (MRR) to evaluate the ranking efficiency of the term extraction metrics. The mean reciprocal rank is calculated by averaging the ranks of the retrieved candidates against all possible candidates.

	Accuracy (Top1)	Accuracy (Top5)	MRR
LM-PMI	28.29	45.18	1.632
C-Value	23.26	32.26	2.263

Table 1: Accuracy and Mean Reciprocal Rank for Term Extracted from the WikiFood Corpus

Table 1 presents the results of the experiment on term extraction for the WikiFood Corpus. The LM-PMI value clearly scores better in terms of accuracy to rank the correct term candidate in the top position with a mean rank of 1.63 and an accuracy of 28.29 compared to the C-values mean rank of 2.26 and an accuracy of 23.26.

3 Ontology Induction

Manually created ontologies such as CYC (Lenat, 1995) and WordNet (Miller, 1995)) are resource and time intensive and they also suffer from coverage sparsity. This motivated to move towards unsupervised approaches for ontology induction and knowledge extraction (Lin and Pantel, 2001; Snow et al., 2006; Velardi et al., 2013). Ontological induction approaches can be broadly categorized as (i) pattern/rule based, (ii) clustering based, (iii) graph based and (iv) vector space approaches.

3.1 Rule-based, Clustering and Graph-based Approaches

The first notable rule-based ontology learning approach exploits lexico-syntactic patterns that explicitly links a hypernym to its hyponym, e.g. "*X and other Ys*" and "*Ys such as X*". These patterns are either manually constructed (Berland and Charniak, 1999; Kozareva et al., 2008) or automatically bootstrapped (Girju, 2003). They rely on surface-level patterns produce many false positive terms due to parsing error and polysemy.

Clustering based approaches are usually used to discover hypernymy and synonym relations. For example, Lin (1998) clustered similar words based on binary comparisons of the amount of information needed to differentiate the distributional commonality between two words.

Different from most hypernymy clustering-based induction methods focus on a bottom-up approach, (Caraballo, 2001; Lin, 1998), Pantel and Ravichandran (2004) introduced a top-down approach, assigning the hypernyms to clusters using co-occurrence statistics and then pruning the cluster by recalculating the pairwise similarity between every hyponym pair within the cluster.

Contrary to the hierarchical top-up/bottom-down approach to ontological induction, graph-based algorithms produce an open-ended network graph by connecting terms through directed relations. In this regard, a tree-like structure is not guaranteed unless acyclicity is assured during the graph building process.

3.2 Vector Space Approaches

Although vector space models have been used widely in other NLP tasks, ontology/taxonomy inducing using vector space models has not been popular. It is only since the recent advancement in neural nets and word embeddings that vector space models are gaining ground for ontology induction and relation extraction (Saxe et al., 2013; Khashabi, 2013).

In a vector space approaches, lexical semantic knowledge can be thought of as a two-dimensional euclidean space where each word is represented as a point and semantic association is indicated by word proximity. The vector space representation for each word is constructed from the distribution of words across context, such that words with similar meaning are found close to each other in the space (Mitchell and Lapata, 2010; Tan, 2013). Fu et al. (2014) proposed a vector space approach to hypernym-hyponym identification using word embeddings that trains a feature function that converts a hyponym vector to its hypernym. However, their approach requires an existing hypernym-hyponym pairs for training before discovering new pairs.

3.2.1 Supervised Vector-Space Approach: Projecting a Hyponym to its Hypernym with Transition Matrix

Fu et al. (2014) discovered that hypernym-hyponyms pairs have similar semantic properties as the linguistics regularities discussed in Mikolov et al. (2013b). For instance: v(shrimp)-v(prawn) $\approx$ v(fish)-v(goldfish). Intuitively, the assumption is that all words can be projected to their hypernyms based on a transition matrix. That is, given a word x and its hypernym y, a transition matrix Φ exists such that $y = \Phi x$, e.g. v(goldfish) = $\Phi \times v$(fish).

Fu et al. proposed two projection approaches to identify hypernym-hyponym pairs, (i) uniform linear projection where Φ is the same for all words and Φ is learnt by minimizing the mean squared error of $\|\Phi x\text{-}y\|$ across all word-pairs (i.e. a domain independent Φ) and (ii) piecewise linear projection that learns a separate projection for different word clusters (i.e. a domain dependent Φ, where a taxonomy's domain is bounded by its terms' cluster(s)). In both projections, hypernym-hyponym pairs are required to train the transition matrix Φ.

3.2.2 Unsupervised Vector-Space Approach: Inducing a Hypernym with *is-a* Vector

Instead of learning a supervised transition matrix Φ, we propose a simpler unsupervised approach where we learn a vector for the phrase "*is-a*". We single-tokenize the adjacent "is" and "a" tokens and learn the word embeddings with *is-a* forming part of the vocabulary in the input matrix (Tan et al., 2015a).

Effectively, we hypothesize that Φ can be replaced by the "*is-a*" vector. To achieve the piecewise projection effects of Φ, we trained a different deep neural net model for each TaxEval domain and assume that the "*is-a*" scales automatically across domains. For instance, the multiplication of the $v(\texttt{tiramisu})$ and the $v(\texttt{is-a}_{food})$ vectors yields a proxy vector and we consider the top ten word vectors that are most similar to this proxy vector as the possible hypernyms, i.e. $v(\texttt{tiramisu}) \times v(\texttt{is-a}_{food}) \approx v(\texttt{cake})$.

3.3 Evaluation

To evaluate the efficiency of our novel ontology induction approach, we use the SemEval-2015 Taxonomy Extraction Evaluation (TaxEval) task data that addresses taxonomy learning without the term discovery step, i.e. the terms for which to create the taxonomy are given (Bordea et al., 2015). The focus of the task is on the hypernym-hyponym relations, similar to Fountain and Lapata (2012).

In the TaxEval task, taxonomies are evaluated through comparison with gold standard taxonomies. There is no training corpus provided by the organisers of the task and the participating systems are to generate hyper-hyponyms pairs using a list of terms from four different domains, viz. chemicals, equipment, food and science.

The gold standards used in evaluation are the *ChEBI ontology* for the chemical domain (Degtyarenko et al., 2008), the *Material Handling Equipment taxonomy*[2] for the equipment domain, the *Google product taxonomy*[3] for the food domain and the *Taxonomy of Fields and their Different Sub-fields*[4] for the science domain. In addition, all four domains are also evaluated against the sub-hierarchies from the WordNet ontology that subsumes the Suggested Upper Merged Ontology (Pease et al., 2002).

3.3.1 Evaluation Metrics

For the TaxEval task, the multi-faceted evaluation scheme presented in Navigli (2013) was adopted to compare the overall structure of the taxonomy against a gold standard, with an approach used for comparing hierarchical clusters. The multi-faceted evaluation scheme evaluates (i) the structural measures of the induced taxonomy (left columns of Table 1), (ii) the comparison against gold standard taxonomy (right columns of Table 1 and leftmost column of Table 2) and (iii) manual evaluation of novel edges precision (last row of Table 2).

Regarding the two types of automatic evaluation measures, the structural measures provides a gauge of the system's coverage and the ontology structural integrity, i.e. "tree-likeness" of the ontology produced by the hypernym-hyponym pairs, and the comparison against the gold standards gives an objective measure of the "human-likeness" of the system in producing a taxonomy that is similar to the manually-crafted taxonomy.

3.4 Experimental Setup

There is no specified training corpus released for the SemEval-2015 TaxEval task. To produce a domain specific corpus for each of the given domains in the task, we used the Wikipedia dump and preprocessed it using WikiExtractor[5] and then extracted documents that contain the terms for each domain individually.

We trained a skip-gram model phrasal word2vec neural net (Mikolov et al., 2013a) using the `gensim` toolkit (Řehůřek and Sojka, 2010). The neural nets were trained for 100 epochs with a window size of 5 for all words in the corpus.

[2]http://www.ise.ncsu.edu/kay/mhetax/index.htm

[3]http://www.google.com/basepages/producttype/taxonomy.en-US.txt

[4]http://sites.nationalacademies.org/PGA/Resdoc/PGA_044522

[5]We use the same Wikipedia dump to text extraction process from the SeedLing - Human Language Project (Emerson et al., 2014).

3.5 Results

	|V|	|E|	#c.c	cycles	#VC	%VC	#EC	%EC	:NE
Chemical	13785	30392	302	YES	13784	**0.7838**	2427	0.0977	1.1268
Equipment	337	548	28	YES	336	0.549	227	0.3691	0.5219
Food	1118	2692	23	YES	948	0.6092	428	0.2696	1.4265
Science	355	952	14	YES	354	0.7831	173	**0.3720**	1.6752
WN Chemical	1173	3107	31	YES	1172	**0.8675**	532	**0.3835**	1.8566
WN Equipment	354	547	43	YES	353	0.7431	149	0.3072	0.8206
WN Food	1200	3465	23	YES	1199	0.8068	549	0.3581	1.9021
WN Science	307	892	8	YES	306	0.7132	156	0.3537	1.6689

Table 2: Structural Measures and Comparison against Gold Standards for `USAAR-WLV`. The labels of the columns refer to no. of distinct vertices and edges in induced taxonomy (**|V|** and **|E|**), no. of connected components (**#c.c**), whether the taxonomy is a Directed Acyclic Graph (**cycles**), vertex and edge coverage, i.e. proportion of gold standard vertices and edges covered by system (**%VC** and **%EC**), no. of vertices and edges in common with gold standard (**#VC** and **#EC**) and ratio of novel edges (**:NE**).

Table 2 presents the evaluation scores for our system (USAAR-WLV) in the TaxEval task, the %VC and %EC scores summarize the performance of the system in replicating the gold standard taxonomies.

In terms of vertex coverage, our system performs best in the chemical and WordNet chemical domain. Regarding edge coverage, our system achieves highest coverage for the science domain and WordNet chemical domain. Having high edge and vertex coverage significantly lowers false positive rate when evaluating hypernym-hyponyms pairs with precision, recall and F-score.

We also note that the Wikipedia corpus extracted that we used to induce the vectors lacks coverage for the food domain. In the other domains, we discovered all terms in the wikipedia corpus plus the domains' root hypernym (i.e. **|V| = #VC** + 1).

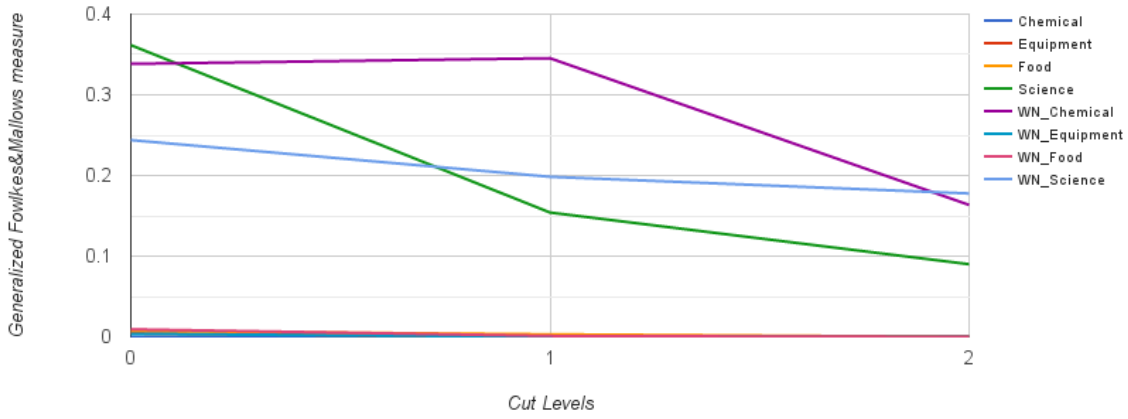


Figure 1: Generalized Cumulative F&M Scores across Domains for Various Cut Levels

Figure 1 shows that our generalized cumulative F&M scores are only valid for a maximum cut-level of 2 for Science, WordNet Chemical and WordNet Science domain, after which the F&M scores falls to 0. In all other domains the F&M scores remains negligibly close to 0 at all cut-levels. While our system's F&M measure is low, it is only representative of the clusters we have induced as compared to the gold standard. To improve our F&M measure, we could reduce the number of redundant novel edges by pruning our system outputs and achieve comparable results to the other teams given our relatively precision of novel edges.

	INRIASAC	LT3	NTNU	QASSIT	TALN-UPF	USAAR-WLV
Avg. F&M	0.3270	**0.4130**	0.0580	0.3880	0.2630	0.0770
Avg. Precision	0.1721	**0.3612**	0.1754	0.1563	0.0720	0.2014
Avg. Recall	0.4279	**0.6307**	0.2756	0.1588	0.1165	0.3139
Avg. F-Score	0.2427	**0.3886**	0.2075	0.1575	0.0798	0.2377
Avg. Precision of NE	0.4800	**0.5960**	0.3530	0.2470	0.1020	0.4200

Table 3: Averaged F&M Measure, Precision, Recall, F-score for All Systems Outputs when Compared to Gold Standard and Manually Evaluated Average Precision of Novel Edges.

Table 3 presents the comparative results between the participating teams in the TaxEval task averaged over all domains. We performed reasonable well as compared to the other systems in all measures, ranking third in among the competitors behind INRIASAC and LT3. The INRIASAC system induced the hyper-hyponym pairs by using frequency-based co-occurrence statistics and substring heuristics (Grefenstette, 2015) while the LT3 induction system uses a combination of rule-based lexical-syntactic patterns and morphological analysis, in addition, the LT3 system incorporated hypernym relations from existing structured lexical resources (Lefever, 2015).

Although the QASSIT system ranked fifth in the task due to the low precision, recall and f-scores, it achieved the highest F&M scores because the system's approach is grounded on graph related theories. The QASSIT system uses a pretopological approach to to model subsumption relations and transforms a list of terms into a structured term space by combining different discriminant criteria (Cleuziou et al., 2015). The pretopology theory generalizes topology and graph theories (Brissaud, 1975; Biggs et al., 1976) and it's commonly used in lexical taxonomy researches to model complex propagation phenomena using a pseudo-closure operator.

4 Conclusion

In this paper, we have proposed a novel Language Model - Pointwise Mutual Information (LM-PMI) method for term extraction using a pre-computed language model and pointwise mutual information between the nested term candidates and a novel method for a vector-space approach to using the 'is-a' vector.

Our preliminary experiments using the LM-PMI term extractor have shown promising results as compared to the terms extracted using the commonly used C-value. The use of a pre-built language model efficiently calculates the logarithmic probabilities of term candidates and allows the assignment of probability to an unknown word, which was previously not possible.

Our unsupervised approach to ontology induction have simplified a previously complex supervised process of inducing a hypernym-hyponym pairs from a neural net by using a non-content phrase vector. Our system achieved modest results when compared against other state-of-art ontology induction system. Given the simple approach to hypernym-hyponym relations, it is possible that future research can apply the same method to other non-content phrase vectors to induce other relations between entities.

5 Ongoing Work

The terminology and ontology research under the EXPERT project thus far has created innovative technologies for term extraction and ontology induction and in both aspects have shown competitive or better performance as compared to state-of-art approaches. The last phase of the project is to apply the new innovations to improve translation quality.

We identify two uses of terminology within the typical translation workflow; (i) as additional domain-specific lexical knowledge for machine translation and (ii) providing domain-specific knowledge in form of a term-bank for human translators/post-editors. To validate the improvement of machine translation quality, we intend to use automatically extracted terms from training corpus to over-weigh the probability mass of the phrases in machine translation by adding extra instances of the terms to the monolingual

language model or bilingually extending the parallel data before the translation model training as implemented in (Tan and Pal, 2014) or (Lefever et al., 2009).

To contribute towards better human translations, we will be extracting terms and automatically classifying/clustering the terms into several technical domains such that they are freely available for human translators in the standard TBX format[6].

As for the application of ontology to improve translation quality, we propose the use of ontologized parallel data instead of surface strings when training a machine translation model. By ontologizing the data, we refer to annotated parallel text that contains unique ontological identification labels that can either be used (i) to resolve lexical ambiguities by making the translation model vocabulary more sparse by replacing surface strings with ontological terms or (ii) as additional annotation factors in factored model machine translation models.

An automatically induced ontology can also be used as an additional factor in machine translation evaluation and quality estimation. It is possible that a hyponym could be translated as a hypernym in their target language and automatically identifying hyper-hyponym pairs across the source and hypothesis translation could improve the semantic similarity component of machine translation evaluation and quality estimation, e.g. Béchara et al. (2015);Arora et al. (2015); Tan et al. (2015b) and Scarton and Specia (2014).

Acknowledgements

Liling Tan is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471.

References

Piyush Arora, Chris Hokamp, Jennifer Foster, and Gareth Jones. 2015. Dcu: Using distributional semantics and domain adaptation for the semantic textual similarity semeval-2015 task 2. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 143–147, Denver, Colorado, June.

Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. 2015. Miniexperts: An svm approach for measuring semantic textual similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 96–101, Denver, Colorado, June.

Matthew Berland and Eugene Charniak. 1999. Finding Parts in Very Large Corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64.

Norman Biggs, E. Keith Lloyd, and Robin J. Wilson. 1976. *Graph theory 1736-1936*. Clarendon Press.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy Extraction Evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

Marcel Brissaud. 1975. *Les espaces pretopologiques*. Compte-rendu de lAcademie des Sciences.

Sharon Ann Caraballo. 2001. *Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text*. Ph.D. thesis, Providence, RI, USA. AAI3006696.

Stanley F Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Guillaume Cleuziou, Davide Buscaldi, Gael Dias, Vincent Levorato, and Christine Largeron. 2015. QASSIT: A Pretopological Framework for the Automatic Construction of Lexical Taxonomies from Raw Texts. In *Proceedings of Ninth International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, USA.

[6]If resources and time permits, the resulting term bases will enhanced with additional terms retrieved from online translation memory and web data using the web-crawler built by Tan and Bond (2011), Tan et al. (2014),Tan and Ordan (2015)

Kirill Degtyarenko, Paula De Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan Mcnaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj, and Michael Ashburner. 2008. ChEBI: A Database and Ontology for Chemical Entities of Biological Interest. *Nucleic acids research*, 36(suppl 1):D344–D350.

Guy Emerson, Liling Tan, Susanne Fertmann, Alexis Palmer, and Michaela Regneri. 2014. SeedLing: Building and Using a Seed corpus for the Human Language Project. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 77–85.

Helmut Flber. 1984. *Terminology Manual*. International Information Centre for Terminology.

Trevor Fountain and Mirella Lapata. 2012. Taxonomy Induction using Hierarchical Random Graphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 466–476.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209.

Roxana Girju. 2003. Automatic Detection of Causal Relations for Question Answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*, pages 76–83.

Gregory Grefenstette. 2015. INRIASAC: Simple Hypernym Extraction Methods. In *Proceedings of Ninth International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, USA.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Andrew Hippisley, David Cheng, and Khurshid Ahmad. 2005. The head-modifier principle and multilingual term extraction. *Natural Language Engineering*, 11(02):129–157.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289.

Daniel Khashabi. 2013. On the Recursive Neural Networks for Relation Extraction and Entity Recognition. Technical report.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio, June.

Els Lefever, Lieve Macken, and Veronique Hoste. 2009. Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 496–504, Athens, Greece, March. Association for Computational Linguistics.

Els Lefever. 2015. LT3: A Multi-modular Approach to Automatic Taxonomy Construction. In *Proceedings of Ninth International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, USA.

Douglas B Lenat. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38.

Dekang Lin and Patrick Pantel. 2001. Discovery of Inference Rules for Question-Answering. *Natural Language Engineering*, 7(04):343–360.

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

George A Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1439.

David S. Palermo and James J. Jenkins. 1964. *Word Association Norms*. University of Minnesota Press.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically Labeling Semantic Classes. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Adam Pease, Ian Niles, and John Li. 2002. The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In *Working Notes of the AAAI-2002 Workshop on Ontologies and the SemanticWeb*, Edmonton, Canada.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2013. Learning Hierarchical Category Structure in Deep Neural Networks. pages 1271–1276.

Carolina Scarton and Lucia Specia. 2014. Exploring consensus in machine translation for quality estimation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 342–347, Baltimore, Maryland, USA, June.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808.

Liling Tan and Francis Bond. 2011. Building and annotating the linguistically diverse ntu-mc (ntu-multilingual corpus). In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 362–371, Singapore, December. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Liling Tan and Noam Ordan. 2015. Usaar-chronos: Crawling the web for temporal annotations. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 846–850, Denver, Colorado, June. Association for Computational Linguistics.

Liling Tan and Santanu Pal. 2014. Manawi: Using multi-word expressions and named entities to improve machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 201–206, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Liling Tan, Anne Schumann, Jose Martinez, and Francis Bond. 2014. Sensible: L2 translation assistance by emulating the manual post-editing process. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 541–545, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Liling Tan, Rohit Gupta, and Josef van Genabith. 2015a. USAAR-WLV: Hypernym Generation with Deep Neural Nets. In *Proceedings of Ninth International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, USA.

Liling Tan, Carolina Scarton, Lucia Specia, and Josef van Genabith. 2015b. Usaar-sheffield: Semantic textual similarity with deep regression and machine translation evaluation metrics. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 85–89, Denver, Colorado, June. Association for Computational Linguistics.

Liling Tan, Josef van Genabith, Marcos Zampieri, Anne Schumann, Jon Dehdari, and Santanu Pal. 2015c. D4. 2: Terminology and ontology. Technical report, EXPERT (EXPloiting Empirical appRoaches to Translation) Consortium.

Liling Tan. 2013. Examining crosslingual word sense disambiguation. Master's thesis, Nanyang Technological University.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A Graph-based Algorithm for Taxonomy Induction. *Computational Linguistics*, 39(3):665–707.

Eugen Wüster. 1969. Die vier Dimensionen der Terminologiearbeit. *Mitteilungsblatt fr Dolmetscher und bersetzer*, 2(15):1–6.

Artificial Data Generation for Quality Estimation

Varvara Logacheva
University of Sheffield,
Sheffield, United Kingdom
v.logacheva@sheffield.ac.uk

Abstract

The modelling of natural language tasks using data-driven methods is often hindered by the problem of insufficient naturally occurring examples of certain linguistic constructs. The task we address in this paper – quality estimation (QE) of machine translation – suffers from lack of negative examples at training time, i.e., examples of low quality translation. This is particularly true for state of the art translation systems built for closely related languages. We propose various ways to artificially generate examples of translations containing errors and evaluate the influence of these examples on the performance of QE models both at sentence and word levels.

1 Introduction

The task of classifying texts as "correct" or "incorrect" often faces the problem of unbalanced training sets: examples of the "incorrect" class can be very limited or even absent. In many cases, naturally occurring instances of these examples are rare (e.g. incoherent sentences, errors in human texts). In others, the labelling of data is a non-trivial task which requires expert knowledge.

Consider the task of quality estimation (QE) of machine translation (MT) systems output. When performing binary classification of automatically translated sentences one should provide examples of both bad and good quality sentences. Good quality sentences can be taken from any parallel corpus of human translations, whereas there are very few corpora of sentences annotated as having low quality. These corpora need to be created by human translators, who post-edit automatic translations, mark errors in translations, or rate translations for quality. This process is slow and expensive. It is therefore desirable to devise automatic procedures to generate negative training data for QE model learning.

Previous work has followed the hypothesis that machine translations can be assumed to have low quality (Gamon et al., 2005). However, this is not the case nowadays: many translations can be considered flawless. Particularly for word-level QE, it is unrealistic to presume that every single word in the MT output is incorrect. Another possibility is to use automatic quality evaluation metrics based on reference translations to provide a quality score for MT data. Metrics such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Banerjee and Lavie, 2005) can be used to compare the automatic and reference translations. However, these scores can be very unreliable, especially for word-level QE, as every word that differs in form or position would be annotated as bad, although it can be acceptable.

Previous efforts have been made for negative data generation, including random generation of sentences from word distributions and the use of translations in low-ranked positions in n-best lists produced by statistical MT (SMT) systems as the examples of low quality sentences. These methods are however unsuitable for QE at the word level, as they provide no information about the quality of individual words in a sentence.

In this paper we adopt a different strategy: we insert errors in otherwise correct sentences. This provides control over the proportion of errors in the negative data, as well as knowledge about the quality of individual words in the generated sentences. The goals of the research presented here are to understand the influence of artificially generated data (by various methods and in various quantities) on the performance of QE models at both sentence and word levels, and ultimately improve upon baseline

models by extending the training data with suitable artificially created examples. In Section 2 we further review existing strategies for artificial data generation. We explain our generation strategies in Section 3. In Section 4 we describe our experiment and their results.

2 Previous work

Many text generation tasks evaluate a generated utterance with a probability distribution computed on a set of well-formed texts: the more similar to the training data, the better. However, some tasks need to explicitly define which outputs are good and which are bad and these tasks usually lack the examples of erroneous sentences or texts.

2.1 Discriminative language modelling

One example of task that requires low quality examples is discriminative language modelling (DLM), i.e., the classification of sentences as "good" or "bad". It was first introduced in a monolingual context within automatic speech recognition (Collins et al., 2005), and later applied to MT. While in speech recognition negative examples can be created from system outputs that differ from the reference (Bhanuprasad and Svenson, 2008), in MT there are multiple correct outputs, so negative examples need to be defined more carefully.

In Okanohara (2007) bad sentences used as negative training instances are drawn from the distribution $P(w_i|w_{i-N+1}, ..., w_{i-1})$: first the start symbol $< s >$ is generated, then the next words are taken based on the word probability given the already generated words.

Other approaches to discriminative LMs use the n-best list of the MT system as training data (Li and Khudanpur, 2008). The translation variant which is closest to the oracle (e.g. has the highest BLEU score) is used as a positive example, while the variant with high system score and low BLEU score is used as a negative example. Such dataset allows the classifier to reduce the differences between the model score and the actual quality score of a sentence.

Li et al. (2010) simulate the generation of an n-best list using translation tables from SMT systems. By taking entries from the translation table with the same source side they create a set of alternative translations for a given target phrase. For each sentence, these are combined, generating a confusion set for this sentence.

2.2 Quality estimation for MT

QE can be modelled as a classification task where the goal is to distinguish good from bad translations, or to provide a quality score to each translation. Therefore, examples of bad sentences or words produced by the MT system are needed. To the best of our knowledge, the only previous work on adding errors to well-formed sentences is that by Raybaud et al. (2011).

In (Raybaud et al., 2011), the training data for the negative data generation process consists of a set of MT hypotheses manually post-edited by a translator. Hypotheses are aligned with the corresponding post-editions using the TERp tool (Snover et al., 2008). The alignment identifies the edit operations performed on the hypothesis in order to convert it to the post-edited version: leave word as is (no error), delete word, insert new word, substitute word with another word. Two models of generation of error strings from a well-formed sentence are proposed. Both are based on the observed frequency of errors in the post-edited corpus and do not account for any relationships between the errors and the actual words. The *bigram error model* draws errors from the bigram probabilities $P(C_i|C_{i-1})$ where C_i is an error class. The *cluster error model* generates clusters of errors based on the distribution of lengths of erroneous word sequences in the training data. Substituting words are chosen from a probability distribution defined as the product of these words' probabilities in the IBM-1 model and a 5-gram LM. A model trained only on artificial data performs slightly better than one trained on a small manually annotated corpus.

2.3 Human error correction

Another task that can benefit from artificially generated examples is language learner error correction. The input for this task is text that potentially contains errors. The goal is to find these errors, similarly to

QE at the word level, and additionally correct them. While the text is written by humans, it is assumed that these are non-native speakers, who possibly translate the text from their native language. The difference is that in this task the source text is a hidden variable, whereas in MT it is observed.

The strategy of adding errors to correct sentences has also been used for this task. Human errors are more intuitive to simulate as language learners explicitly attempt to use natural language grammars. Therefore, rule-based systems can be used to model some grammar errors, particularly those affecting closed class words, e.g. determiner errors (Izumi et al., 2003) or countability errors (Brockett et al., 2006).

More recent statistical methods use the distributions of errors in corpora and small seed sets of errors. They often also concentrate on a single error type, usually with closed class words such as articles and prepositions (Rozovskaya and Roth, 2010). Felice and Yuan (2014) go beyond closed class words to evaluate how errors of different types are influenced by various linguistic parameters: text domain, learner's first language, POS tags and semantic classes of erroneous words. The approach led to the generation of high-quality artificial data for human error correction. However, it could not be used for MT error identification, as MT errors are different from human errors and usually cannot be assigned to a single type.

3 Generation of artificial data

The easiest choice for artificial data generation is to create a sentence by taking all or some of its words from a probability distribution of words in some monolingual corpus. The probability can be defined for unigrams only or conditioned on the previous words (as it was done for discriminative LMs). This however is a target language-only method that does not suit the QE task as the "quality" of a target word or sentence is dependent on the source sentence, and disregarding it will certainly lead to generation of spurious data.

Random target sentences based on a given source sentence could be generated with bilingual LMs. However another limitation of this approach is the assumption that all words in such sentences are wrong, which makes the data useless for word-level QE.

Alternatively, the artificial sentences can be generated using MT systems for back-translation. The target sentences are first fed to a target–source MT system, and then its output is passed to a source–target system. Back-translations are more similar to the original sentence than to an arbitrary human reference. However, according to our experiments, if both systems are statistical the back-translation is too similar to the original sentence, and the majority of their differences are interchangeable paraphrases. Rule-based systems could be more effective, but the number of rule-based systems freely available would limit the work to a small number of language pairs.

3.1 A two-stage error generation method

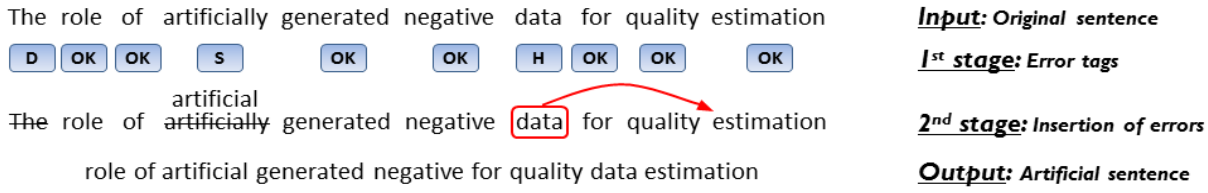


Figure 1: Example of the two-stage artificial data generation process

As previously discussed, existing methods that artificially generate entire sentences have drawbacks that make them difficult or impossible to use for QE. Therefore, following Raybaud et al. (2011) and previous work on human error correction, our approach is to inject errors into otherwise correct texts. This process consists of two stages:

- labelling of a sentence with error tags,

- insertion of the errors into that sentence.

The first stage assigns an error tag to every word in a sentence. The output of this stage is the initial sentence where every word is assigned a tag denoting a type of error that needs to be incurred on this word. We use five tags corresponding to edit operations in the TERp tool: no error (**OK**), substitution (**S**), deletion (**D**), insertion (**I**) and shift (**H**). During the second stage the words in the sentence are changed according to their tag: substituted, deleted, shifted, or left in place if word has the tag **OK**. Figure 1 gives an example of the complete generation process.

3.1.1 Error tagging of sentences

We generate errors based on a corpus of post-edited machine translations. We align translations and post-editions using the TERp tool (exact matching) and extract counts on the number of shifts, substitutions, insertions and deletions. TERp does not always capture the true errors, in particular, it fails to identify phrase substitutions (e.g. *was* → *has been*). However, since editors are usually asked to minimise the number of edits, translations and post-editions are often close enough and the TERp alignment provide a good proxy to the true error distribution.

The TERp alignments can be used to collect the statistics on errors alone or to combine the frequency of errors with the words they are incurred on. We suggest three methods of generation of an error string for a sentence:

- **bigramEG**: the *bigram* error generation that uses a bigram error model regardless of the actual words (Raybaud et al., 2011).
- **wordprobEG**: the conditional probability of an error given a word.
- **crfEG**: the combination of the bigram error model and error probability conditioned on a word. This generation method can be modelled with Hidden Markov Model (HMM) or conditional random fields (CRF).

The first model has the advantage of keeping the distribution of errors as in the training data, because the probability distributions used depend only on the frequency of errors themselves. The second model is more informed about which words commonly cause errors. Our implementation of the third method uses CRFs to train an error model. We use all unigrams, bigrams and trigrams that include the target word as features for training. This method is expected to produce more plausible error tags, but it can have the issue that the vocabulary we want to tag is not fully covered by the training data, so some words in the sentences to tag will be unknown to the trained model. If an unknown word needs to be tagged, it will more often be tagged with the most frequent tag, which is "Good" in our case. In order to avoid this problem we replace rare words in the training set with a default string or with the word class, i.e. the word's POS tag. We also consider the scenario where the POS tags are used as additional features.

The training data needed for this stage of data generation is a corpus of well-formed target-language sentences where each word is tagged with a tag corresponding to an error which is likely to be made in this word during machine translation (we use the same five error tags: 'OK' for correct word, 'I' for insertion of word, 'D' for deletion of word, 'S' for substituting the word with another (incorrect) word, and 'H' for shifting word to another position). This corpus is achieved from a collection of automatic translations with post-editions. We align automatic translations with their post-editions with TERp tool, and see all the differences between them, and the type of these differences (inserted word, changed word, etc.). We take the post-editions with corresponding error tags as the training data for error generators.

3.1.2 Insertion of errors

We consider errors of four types: **insertion**, **deletion**, **substitution** and **shift**. Shift errors require the distribution of shift distances which are computed based on the TERp-aligned training corpus. Substitutions and insertions need the new words to be drawn from some list of words with a probability distribution assigned to it, so that every word occurs with some probabilities and probabilities of all words from the list sum to 1.

We suggest three methods for the generation of these lists and distributions:

- **unigramWI**: common list for all the words. The word list is a vocabulary of some large monolingual corpus, the probabilities are frequencies of words in this corpus.

- **paraphraseWI**: separate list for every word (including a special list for out-of-vocabulary words). Every word is assigned a list of possible paraphrases. The paraphrases for a word are defined as follows. First all possible sources of a target word are extracted from an SMT system's lexical translation table. Then the reverse lexical translation table is used to extract all target words that can be translations of these sources (see table 1). The probabilities of the paraphrases are computed as $P(w') = P(s|w) \times P(w'|s)$, where w is the considered word, and w' are words from its paraphrase list. The distribution $P(w')$ should be normalised so that it sums to 1. That gives us a confusion set for each target word.

- **lexprobWI**: separate list for every **source** word, which contains the possible translations of the word. This method is similar to the previous one (see table 2): the lexical translation table is searched for the word pairs where the source side matches the word under consideration, all target sides of the found pairs are added to the translation list.

target	source	⇒	source	**target**	→	**Paraphrases list**
target	source	⇒	source	**target**	→	
target	source	⇒	source	**target**	→	
target	source	⇒	source	target		
target	source	⇒	source	**target**	→	
target	source	⇒	source	target		
target	source	⇒	source	**target**	→	
target	source	⇒	source	target		
target	source	⇒	source	**target**	→	

Table 1: Generation of a paraphrase list for **paraphraseWI**. Same colours denote same words.

source	target		**Translations list**
source	**target**	→	
source	**target**	→	
source	target		
source	**target**	→	
source	target		
source	**target**	→	
source	target		
source	**target**	→	

Table 2: Generation of a translations list for **lexprobWI**. Same colours denote same words.

The **lexprobWI** is different from other word inserters: it does not fit into the data generation scenario described above. While **unigramWI** and **paraphraseWI** perform changes to the sentence in the *target* language, the **lexprobWI** needs a *source* sentence. However, the word inserters take their input from the error generators which assign error tags to words of a valid sentence. Therefore, in order to be compatible with **lexprobWI**, error generators need to assign errors to source sentences.

We trained another set of error generators which tags source sentences with errors. That required a different training set: instead of target sentences with error markup we needed analogous source-language data. Similarly to the training data preparation procedure described in 3.1.1, we align automatic translations with post-editions using TERp to achieve the error markup. After that, we align the post-editions with the corresponding source sentences using one of the alignment tools used in MT systems, e.g. GIZA++ (Och and Ney, 2003), and map the error markup to the source side. Thus, we get a source

corpus where each word is tagged with an error which can be made by an MT system while translating this word into the target language.

4 Experiments

We conducted a set of experiments to evaluate the performance of artificially generated data on different tasks of QE at the sentence and word levels.

4.1 Tools and datasets

The tools and resources required for our experiments are: a QE toolkit to build QE models, the training data for them, the data to extract statistics for the generation of additional examples.

For the sentence-level QE we used the QUEST toolkit (Specia et al., 2013). It trains QE models using `sklearn`[1] versions of Support Vector Machine (SVM) classifier (for ternary classification task, Section 4.4) and SVM regression (for HTER prediction, Section 4.5). The word-level version of QUEST[2] was used for word-level feature extraction. Word-level classifiers were trained with **CRFSuite**[3]. The CRF error models were trained with **CRF++**[4]. POS tagging was performed with **TreeTagger** (Schmid, 1994). Sentence-level QuEst uses 17 baseline features[5] for all tasks. Word-level QuEst reimplements the set of 30 baseline features described in (Luong et al., 2014). The QE models were built and tested based on the data provided for the WMT14 English–Spanish QE shared task (Section 4.3).

The statistics on error distributions were computed using the English–Spanish part of training data for WMT13 shared task on QE[6]. The statistics on the distributions of words, alignments and lexical probabilities were extracted from the Europarl corpus (Koehn, 2005). We trained the alignment model with **FastAlign** (Dyer et al., 2013) and extracted the lexical probability tables for words using scripts for phrase table building in **Moses** (Koehn et al., 2007). For all the methods, errors were injected into the News Commentary corpus[7].

4.2 Generated data

Combining three methods of errors generation and two methods of errors insertion into sentences resulted in a total of six artificial datasets. Here we perform some analysis on the generated data.

The datasets differ in the percentage of errors injected into the sentences (see table 3). The main differences are between the type of error generator used and the language of dataset where the errors were injected.

The datasets where errors were injected into source sentences have significantly lower percentage of errors. This is explained by the fact that the number of error tags is reduced twice by the cross-lingual alignment procedure. First original error tags from the corpus of post-editions are mapped to the source side. Since the alignment procedure does not necessarily produce alignments for every word, some error tags are lost. Then, after assigning tags to source sentences, the **lexprobWI** maps them back to target sentences losing some tags again. This drawback led to low number of errors in the artificial datasets, especially those that used CRF-based error generators.

BigramEG datasets have 23% of edits for the target sentences and 12% for the source sentences, which matches the distribution of errors on the real data. **WordprobEG** datasets contain fewer errors for the target sentences and more errors for the source sentences. This discrepancy The **crfEG** models contain the lowest number of errors when applying them to both source and target sentences. As it was expected, data sparsity makes the CRF model tag the majority of the words with the most frequent tag ("Good"). Replacing rare words with a default word token or with a POS tag did not improve these statistics.

[1] `http://scikit-learn.org/`
[2] `http://github.com/ghpaetzold/quest`
[3] `http://www.chokkan.org/software/crfsuite/`
[4] `https://code.google.com/p/crfpp/`
[5] `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17`
[6] `http://www.quest.dcs.shef.ac.uk/wmt13_qe.html`
[7] `http://statmt.org/wmt14/training-parallel-nc-v9.tgz`

Word inserters	Target language (UnigramWI & ParaphraseWI)	Source language (LexprobWI)
Error generators		
BigramEG	23%	12%
WordprobEG	17%	15%
crfEG	5%	0.7%

Table 3: Percentage of errors in the artificial datasets

Word inserters	UnigramWI	LexprobWI	ParaphraseWI
Error generators			
BigramEG	699.9	285.45	888.64
WordprobEG	538.84	357.6	673.61
crfEG + default word	165.36	137.57	172.97
crfEG + POS tag	161.59	139.72	167.23

Table 4: Perplexities of the artificial datasets

We computed the perplexity of all datasets with respect to an LM trained on the Spanish part of the Europarl corpus (see Table 4). The figures match the error percentages in the data — the lower the number of errors, the more is kept from the original sentence, and thus the more natural it looks (lower perplexity). Note that sentences where errors were inserted from a general distribution (**unigramWI**) have lower perplexity than those generated using using paraphrases. This can be because the **unigramWI** model tends to choose high-frequency words with lower perplexity, while the constructed paraphrases contain more noise and rare words.

4.3 Experimental setup

We evaluated the performance of the artificially generated data in three tasks: the ternary classification of sentences as "good", "almost good" or "bad", the prediction of HTER (Snover et al., 2009) score for a sentence, and the classification of words in a sentence as "good" or "bad" (tasks 1.1, 1.2 and 2 of WMT14 QE shared task[8], respectively).

The goal of the experiments was to check whether it is possible to improve upon the baseline results by adding artificially generated examples to the training sets. The baseline models for all tasks were trained on the data provided for the corresponding shared tasks for the English–Spanish language pair. All models were tested on the official test sets provided for the corresponding shared tasks.

Since we know how many errors were injected into the sentences, we know the TER scores for our artificial data. The discrete labels for the ternary classification task are defined as follows: "bad" sentences have four or more non-adjacent errors (two adjacent erroneous words are considered one error), "almost good" sentences contain one erroneous phrase (possibly of several words), and "good" sentences are error-free. For the sentence ternary classification task we added only "bad" artificially generated sentences to the training sets, for the rest of the tasks we used all the generated sentences.

The new training examples were added to the baseline datasets. We ran a number of experiments gradually increasing the number of artificially generated sentences used. At every run, the new data was chosen randomly in order to reduce the influence of outliers. In order to make the results more stable, we ran each experiment 10 times and averaged the evaluation scores.

4.4 Sentence-level ternary QE task

The original dataset for this task contains 949 "good", 2010 "almost good", and 857 "bad" sentences, whereas the test set has 600 entries: 131 "good", 333 "almost good", 136 "bad". The results were evaluated using F1-score.

[8] http://statmt.org/wmt14/quality-estimation-task.html

The addition of new "bad" sentences leads to an improvement in quality, regardless of the sentence generation method used. Models trained on datasets generated by different strategies display the same trend: adding up to 400 sentences results in a considerable increase in quality, while further addition of data only slightly improves quality.

The best-performing error generator is **crfEG**, however, **bigramEG** performs very closely. Figure 2 shows the results of the experiments – here for clarity we included only the results for datasets generated with the **unigramWI**. The best F1-score of 0.49 is achieved by a model trained on the data generated with the **crf** error generator, which is an absolute improvement of 1.9% over the baseline.

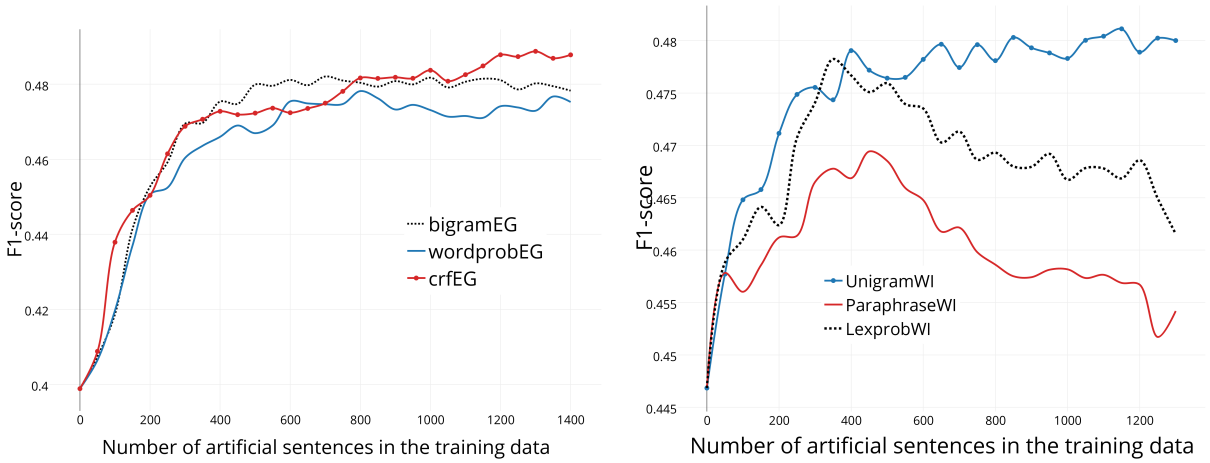


Figure 2: Ternary classification: performance of different error generators (left) and word inserters (right)

Surprisingly, the best word insertion strategy is the simplest one, namely the **unigramWI**, which chooses words according to their frequency in a corpus. The two strategies based on lexical translation probabilities perform worse. This can be explained by the fact that random selection tends to choose frequent words more often, whereas lexical translation tables contain many low-probability translations and even noise. Therefore, the words selected from them often do not suit the sentences and only hamper the system performance. The **paraphraseWI** performs even worse than **lexprobWI**, because it takes the data from two lexical translation tables, which increases the probability of adding noise to the data. The comparison of word insertion strategies is plotted in the figure 2.

However, adding only negative data makes the distribution of classes in the training data less similar to that of the test set, which might affect performance negatively. Therefore, we conducted other three sets of experiments: we added (i) equal amount of artificial data for the "good" and "bad" classes (ii) batches of artificial data for all classes that keep the original proportion of classes in the data (iii) artificial data for only the "good" class. The latter setting is tested in order to check whether the classifier benefits from negative instances, or just from having new data added to the training sets. The "good" sentences were taken from the corpus used for data generation, and "almost good" sentences were naturally generated by our generation methods.

The results are shown in Figure 3. We plot only the results for the **bigramEG + unigramWI** setting as it achieved the best result in absolute values, but the trends are the same for all data generation techniques. The best strategy was to add both "good" and "bad" sentences: it beats the models which uses only negative examples, but after 1000 artificial sentences its performance degrades. Keeping the original distribution of classes is not beneficial for this task: it performs worse than any other tested scenario since it decreases the F1-score for the "good" class dramatically.

Overall, the additional negative training data improves the ternary sentence classification. The addition of both positive and negative examples can further improve the results, while providing additional instances of the "almost good" class did not seem to be as helpful.

Also, as it was already discussed, the datasets formed by **lexprobWI** have less injected errors. It means that we need to generate more data in order to get the sufficient number of artificial negative examples.

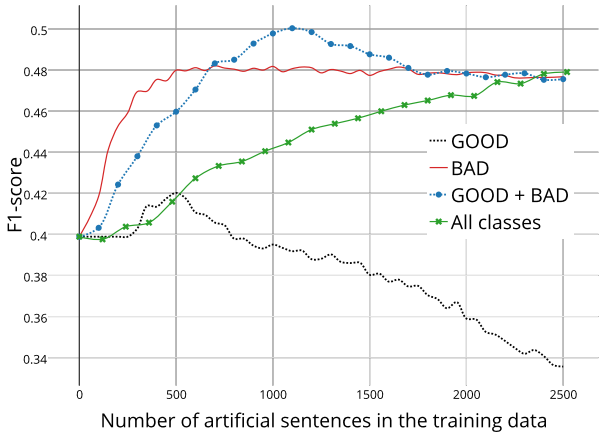


Figure 3: Ternary classification: artificial examples of different classes

In this experiment, we could not test the **crfEG + lexprobWI** combination, because the number of errors in datasets generated in this setting was too small, and we were not able to find the sufficient number of sentences which could be classified as "bad" (i.e. those containing 4 or more errors).

This problem also held for all datasets that used **crfEG** in conjunction with other word inserters: only 3–4% of data generated with CRF-based methods suited for the task. Hence, although CRF-based methods are slightly better for generating the negative data for this task, the fact that they insert too few errors makes them impractical.

4.5 Sentence-level HTER QE task

The prediction of HTER can be more naturally modelled as a regression task, so it was performed using the SVM regression. The results were evaluated in terms of Mean Absolute Error (MAE).

The addition of any type of artificial data leads to substantial improvements in quality for this task. The initial training dataset was very small – 896 sentences (200 sentences for test), which may explain the substantial improvements in prediction quality as new data is added.

The performance of systems depends both on error generators and word inserters used. When using **bigramEG** and **wordprobEG** we noticed, unlike the results of ternary classification task, that the **lexprobWI** is the best-performing word inserter. **UnigramWI** is second best, and **paraphraseWI** has the lowest performance. However, this does not hold for **crfEG** — its combinations with all word inserters create datasets of similar quality (see figure 4).

Therefore, the best strategy of word selection for this task is **lexprobWI**. We compare different error generators in conjunction with **lexprobWI** in figure 5. The addition of data from datasets generated with **crfEG** gives the largest drop in MAE (from 0.161 to 0.138). This result is achieved by a model that uses 750 artificial sentences. Further addition of new data harms performance. The data generated by other error generators does not cause such a sharp improvement, however, it results in steady reduction of error and performs better than **crfEG** as we add more than 1500 artificial sentences.

As it was described earlier, the **crfEG** and **lexprobWI** models generate sentences with a small number of errors. Since the use of datasets generated with these techniques leads to the largest improvements, we can suggest that in the HTER prediction task, using the baseline dataset only, the majority of errors is found in sentences whose HTER score is low. However, the reason might also be that the distributions of scores in the baseline training and test sets are different: the test set has lower average score (0.26 compared to 0.31 in the training set) and lower variance (0.03 versus 0.05 in the training set). The use of artificial data with a small number of errors changes this distribution.

We also experimented with training a model using only artificial data. The results of models trained on only 100 artificial sentences for each generation method were surprisingly good: their MAE ranged from 0.149 to 0.158 (compared to the baseline result of 0.161 on the original data). However, the further

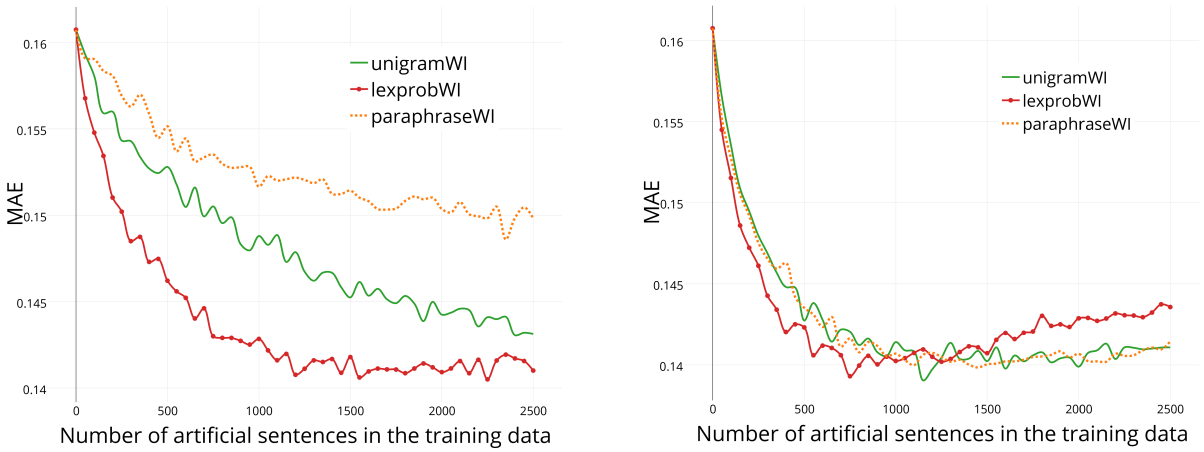


Figure 4: HTER: performance of different word inserters with bigramEG (left) and crfEG (right)

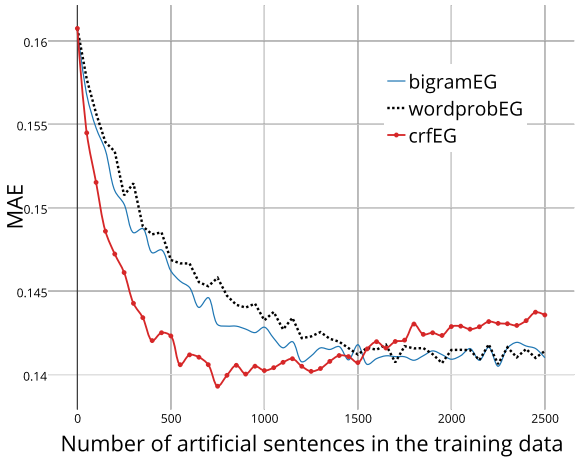


Figure 5: HTER: best-performing datasets (different error generators + lexprobWI)

addition of new artificial sentences did not lead to improvements. Thus, despite the positive impact of the artificial data on the results, the models cannot be further improved without real training examples.

4.6 Word-level QE task

Here we tested the impact of the artificial data on the task of classifying individual words as "good" or "bad". The baseline set contains 47335 words, 35% of which have the tag "bad". The test set has 9613 words with the same label distribution.

All the datasets led to similar results. Overall, the addition of artificial data harms prediction performance: the F1-score goes down until 1500 sentences are added, and then levels off. The performance for all datasets is similar. However, analogously to the previous tasks, there are differences between **crfEG** and the other two error generation techniques: the former leads to faster deterioration of F1-score. No differences were observed among the word insertion techniques tested.

Figure 6 shows the average weighted F1-score and F1-scores for both classes. Since all datasets behave similarly, we show the results for two of them that demonstrate slightly different performance: **crfEG+unigramWI** is shown with solid blue lines, **bigramEG+unigramWI** — with dotted red lines. The use of data generated with CRF-based methods results in slightly faster decline in performance than the use of data generated with **bigramEG** or **wordprobEG**. One possible reason is that the CRF-generated datasets have fewer errors, hence they have different distributions than the original tags in the training data. Therefore, test instances are tagged as "bad" less often. That explains why the F1-score of the "bad" class decreases, whereas the F1-score of the "good" class stays at the same.

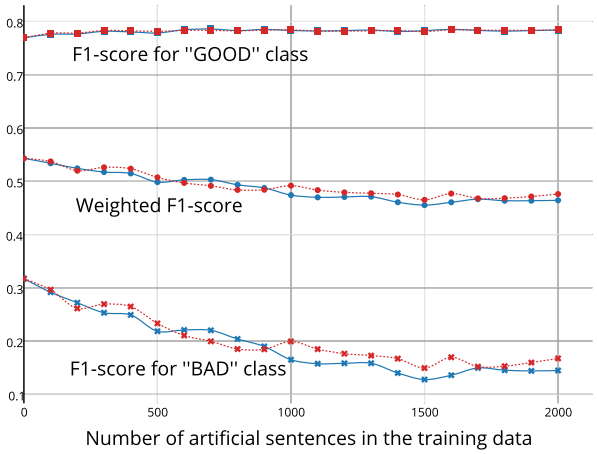


Figure 6: Word-level QE. Blue solid lines – results for **crfEG**, red dotted lines – **bigramEG**

To summarise our findings for word-level QE, the strategies of data generation proposed and tested thus far do not lead to improvements. The word-level predictions are more sensitive to individual words in training sentences, so the replacement of tokens with random words may confuse the model. Therefore, the word-level task needs more elaborate methods for substituting words.

5 Conclusions and future work

We presented and experimented with a set of new methods of simulation of errors made by MT systems. Sentences with artificially added errors were used as training data in models that predict the quality of sentences or words.

The addition of artificial data can help improve the output of sentence-level QE models, with substantial improvements in HTER score prediction and some improvements in sentences classification into "good", "almost good" and "bad". However, the largest improvements are related to the fact that the additional data changes the overall distribution of scores in the training set, making it more similar to the test set. On the other hand, the fact that the artificial sentences did not decrease the quality in such cases proves that it can be used to counter-balance the large number of positive examples. Unlike sentence-level QE, the task of word-level QE did not benefit from the artificial data. That may relate to our choice of method to replace words in artificial sentences.

One of the limitations of our current approach is that the CRF models failed to generate sentences with the sufficient number of errors. To avoid that, the model can be enriched with new features, the training error-labelled sentences could be filtered to include only examples with enough big number of errors, or another sequence labelling model (e.g. HMM) might be more appropriate for the task. In our future research we are planning to upgrade our error generation models based on sequence labelling.

Another problem is the failure of our methods to model phrase substitutions: each word is substituted independently of others, whereas several adjacent errors have a high probability to be related. The future work will include modelling of phrase substitutions.

Acknowledgements

Varvara Logacheva is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL-2005, MTSumm workshop*, pages 65–72.

Kamadev Bhanuprasad and Mats Svenson. 2008. Errgrams A Way to Improving ASR for Highly Inflected Dravidian Languages. In *IJCNLP-2008*, pages 805–810.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Coling-ACL-2006*.

Michael Collins, Brian Roark, and Murat Saraclar. 2005. Discriminative Syntactic Language Modeling for Speech Recognition. In *ACL-2005*.

Chris Dyer, Victor Chahuneau, and A. Noah Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL-HLT-2013*, pages 644–648.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *EACL-2014*, pages 116–126.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level mt evaluation without reference translations: beyond language modeling. In *EAMT-2005*.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the japanese learners' english spoken data. In *ACL-2003*, pages 145–148.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL-2007, Demo session*, pages 177–180.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT-Summit 2005*, pages 79–86.

Zhifei Li and Sanjeev Khudanpur. 2008. Large-scale Discriminative n -gram Language Models for Statistical Machine Translation. In *AMTA-2008*, pages 21–25.

Zhifei Li, Ziyuan Wang, Sanjeev Khudanpur, and Jason Eisner. 2010. Unsupervised Discriminative Language Model Training for Machine Translation using Simulated Confusion Sets. In *Coling-2010*.

Ngoc Quang Luong, Laurent Besacier, and Benjamin Lecouteux. 2014. Lig system for word level qe task at wmt14. In *WMT-2014*, pages 335–341.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Daisuke Okanohara. 2007. A Discriminative Language Model with Pseudo-Negative Samples. In *ACL-2007*, pages 73–80.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL-2002*, pages 311–318.

Sylvain Raybaud, David Langlois, and Kamel Smaïli. 2011. This sentence is wrong. Detecting errors in machine-translated sentences. *Machine Translation*, 25(1):1–34.

Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *EMNLP-2010*, pages 961–970.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *AMTA-2006*, pages 223–231.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2008. TERp System Description. In *AMTA-2008, MetricsMATR workshop*.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter?: Exploring different human judgments with a tunable mt metric. In *WMT-2009*, pages 259–268.

Lucia Specia, Kashif Shah, Jose G C de Souza, and Trevor Cohn. 2013. QuEst - A translation quality estimation framework. In *ACL-2013, Demo session*.

Finding Ways to Assess Machine Translated Documents for Document-level Quality Prediction

Carolina Scarton
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello, Sheffield, S1 4DP, UK
c.scarton@sheffield.ac.uk

Abstract

In this paper we present a framework for document-level quality estimation. The challenges of building such framework, focusing on ideal quality labels for this task are discussed. We also analyse the use of popular automatic machine translation evaluation metrics to provide labels for quality estimation at document and paragraph levels. The crucial limitations of such metrics for this task are highlighted, mainly the fact that they disregard the discourse structure of the texts. To better understand these limitations, we designed experiments with human annotators and proposed a way of quantifying differences in translation quality that can only be observed when sentences are judged in the context of entire documents or paragraphs. Our results indicate that the use of context can lead to more informative labels for quality annotation beyond sentence level. Finally, we propose ways to assess documents, presenting the structure of a large-scale data annotation, using a two-step pos-editions method, in order to move towards document-level prediction using adequate document-level labels.

1 Introduction

Evaluation metrics for Machine Translation (MT) and Automatic Summarisation (AS) tasks should be able to measure quality with respect to different aspects (e.g. fluency and adequacy) and they should be fast and scalable. Human evaluation seems to be the most reliable (although it might introduce biases of reviewers). However, it is expensive and cumbersome for large datasets; it is also not practical for certain scenarios, such as *gisting* in MT and summarisation of webpages.

Automatic evaluation metrics (such as BLEU (Papineni et al., 2002) and ROUGE (Lin and Och, 2004)), based on human references, are widely used to evaluate MT and AS outputs. One limitation of these metrics is that if the MT or AS system outputs a translation or summary considerably different from the references, it does not really mean that it is a bad output. Another problem is that these metrics cannot be used in scenarios where the output of the system is to be used directly by end-users, for example a user reading the output of Google Translate[1] for a given news text cannot count on a reference for that translated text.

Quality estimation (QE) of machine translation (MT) (Blatz et al., 2004; Specia et al., 2009) is an area that focuses on predicting the quality of new, unseen machine translation data without relying on human references. This is done by training models using features extracted from source and target texts and, when available, from the MT system, along with a quality label for each instance.

Most current work on QE is done at the sentence level. A popular application of sentence-level QE is to support post-editing of MT (He et al., 2010). As quality labels, Likert scores have been used for post-editing effort, as well as post-editing time and edit distance between the MT output and the final version – HTER (Snover et al., 2006).

There are, however, scenarios where quality prediction beyond sentence level is needed, most notably in cases when automatic translations without post-editing are required. This is the case, for example, of quality prediction for an entire product review translation in order to decide whether or not it can be published as is, so that customers speaking other languages can understand it.

[1] https://translate.google.com/

The quality of a document is often seen as some form of aggregation of the quality of its sentences. We claim, however, that document-level quality assessment should consider more information than sentence-level quality. This includes, for example, the topic and structure of the document and the relationship between its sentences. While certain sentences are considered perfect in isolation, their combination in context may lead to incoherent text. Conversely, while a sentence can be considered poor in isolation, when put in context, it may benefit from information in surrounding sentences, leading to a document that is fit for purpose.

In this work we focus on document-level QE. We present a framework for QE, called QuEst (Specia et al., 2013; Specia et al., 2015), its extensions for document-level QE and discuss about document-level features. Moreover, we focus on finding the ideal quality label for the task. Document-level quality prediction is a rather understudied problem. Recent work has looked into document-level prediction (Scarton and Specia, 2014; Soricut and Echihabi, 2010) using automatic metrics such as BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) as quality labels. However, their results highlighted issues with these metrics for the task at hand: the evaluation of the scores predicted in terms of mean error was inconclusive. In most cases, the prediction model only slightly improves over a simple baseline where the average BLEU or TER score of the training documents is assigned to all test documents.

Other studies have considered document-level information in order to improve, analyse or automatically evaluate MT output (not for QE purposes). Carpuat and Simard (2012) report that MT output is overall consistent in its lexical choices, nearly as consistent as manually translated texts. Meyer and Webber (2013) and Li et al. (2014) show that the translation of connectives differs from humans to MT, and that the presence of explicit connectives correlates with higher HTER values. Guzmán et al. (2014) explore rhetorical structure (RST) trees (Mann and Thompson, 1987) for automatic evaluation of MT into English, outperforming traditional metrics at system-level evaluation.

Thus far, no previous work has investigated ways to provide a global quality score for an entire document that takes into account document structure, without access to reference translations. Previous work on document-level QE use automatic evaluation metrics as quality labels that do not consider document-level structures and are developed for inter-system rather than intra-system evaluation. Also, previous work on evalution of MT does not focus on complete evaluation at document-level.

In this paper, we show that the use of BLEU and other automatic metrics as quality labels do not help to successfully distinguish different quality levels. We discuss the role of document-wide information for document-level quality estimation and present one experiment with human annotators. This experiment consists in a two-pass post-editing experiment, performed in order to measure the difference between corrections made with and without wider contexts (the tow passes are called PE1 and PE2, repectively).[2]

The results of the two-stage post-editing experiment showed significant differences from the post-editing of sentences without context to the second stage where sentences were further corrected in context. This is an indication that certain translation issues can only be solved by relying on wider contexts, which is a crucial information for document-level QE. A manual analysis was conducted to evaluate differences between PE1 and PE2. Although several of the changes were found to be related to style or other non-discourse related phenomena, many discourse related changes were performed that were only possible given the wider context available.

In the remainder of this paper we first present the QuEst framework for document-level QE and related work in Section 2. In Section 3 we discuss the use of BLEU-style metrics for QE at document level. Section 4 describes the experimental set up used in the paper. Section 5 shows the two-pass post-editing experiment and its results. Section 6 discusses the extension of the two-stage post-edition method in a large scale scenario, with more data and professional annotators. In this section we also discuss ways to integrate the two-stage post-edition scores into a quality label for document-level QE. The conclusions and future work are presented in Section 7.

[2]This paper is an extension of the work reported in Scarton et al. (2015).

2 Related work

2.1 QuEst: framework for Quality Estimation

The extension of QuEst at document level was beyond the development of features. The architecture also needed to be changed in order to support the new level. In this section we describe the features implemented so far and the new architecture proposed.[3]

Features Document-level features implemented in QuEst are the adaptation of the 17 baseline features at sentence level [4] plus nine lexical cohesion features.

Lexical cohesion is a discourse phenomenon related to word repetitions and collocation. This phenomenon was explored as features for Readability Assessment Graesser et al. (2004) and MT evaluation Wong and Kit (2012). Following these work, we proposed the first set of features for QE using lexical cohesion (hereafter, LC).

These features are based in words repetitions only. The reason for that is the aim of keeping QE as language independent as possible. Synonyms and other kind of semantic relations require the need of resources like WordNet (Fellbaum, 1998) that are not freely available for several languages. It is worth mentioning that there are initiatives to fill this gap by using parallel data (Owczarzak et al., 2006; Bannard and Callison-Burch, 2005) that should be explored as future work. Besides that, the coverage of these kind of resources vary across languages, and it could influence in the liability of the feature.

- content words repetition in source and target documents
- lemmas repetition in source and target documents
- nouns repetition in source and target documents
- ratio of content words/lemmas/nouns in source and target documents (three features)

Architecture In order to implement document-level feature extraction in QuEst, its architecture needed to be adapted. Whilst sentence- and word-level QE in QuEst use *Sentence* class as their basic class, document-level QE needs to rely on a *Document* class. One of our aims was also to use the already implemented functionalities of QuEst, such as features at sentence level that could be extended to document level. Therefore, the *Document* class contains a list of *paragraphs*, each paragraph being an object of *Paragraph* class. The *Paragraph* class encompass a list of sentences, each sentence being an object of *Sentence* class. In this scenario, a *document* is a set of *paragraphs* that is a set of *sentences*.

This architecture is flexible in the sense that a document can be considered a combination of paragraphs or sentences (via class objects) or not (by concatenating all sentences together). It is also robust to paragraph-level QE (one could only implement features for this level) and paragraph-driven features for document-level (macro unit features, such as defining the purposes of a given paragraph: introduction, background, conclusion, etc).

Figure 1 shows the architecture for document-level feature extraction and prediction. As mentioned previously, additional classes for document and paragraphs were created. Different from word-level QE, document-level QE can use the same ML pipeline as sentence-level.

Another change that needs to be made in QuEst is how it deals with input files. For word- and sentence-level feature extraction, QuEst receives a raw file, with a sentence per line. However, since we need to deal with documents, a more robust file structure should be supported. We, then, propose the use of SGML files as input for document-level feature extraction. This kind of file can contain several documents with paragraphs and sentences mark-ups. The choice of this format is also supported by the fact that WMT shared tasks use it.

[3] Current version of document-level QuEst: `https://github.com/carolscarton/quest/`

[4] `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17`

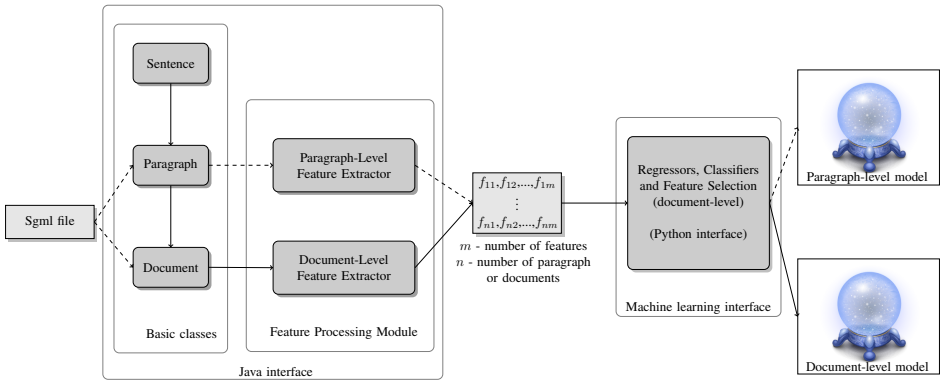


Figure 1: Architecture of QuEst with document-level support. Dashed lines are for work in progress

2.2 Document-level information in MT evaluation

The research reported here is about quality estimation at document-level. Therefore, work on document-level features and document-level quality prediction are both relevant, as well as studies on how discourse phenomena manifest in the output of MT systems.

Soricut and Echihabi (2010) propose document-level features to predict document-level quality for ranking purposes, having BLEU as quality label. While promising results were reported for ranking of translations for different source documents, the results for predicting absolute scores proved inconclusive. For two out of four domains, the prediction model only slightly improves over a baseline where the average BLEU score of the training documents is assigned to all test documents. In other words, most documents have similar BLEU scores, and therefore the training mean is a hard baseline to beat.

Scarton and Specia (2014) propose a number of discourse-informed features in order to predict BLEU and TER at document level. They also found the use of these metrics as quality labels problematic: the error scores of several QE models were very close to that obtained by the training mean baseline. Even when mixing translations from different MT systems, BLEU and TER were not found to be discriminative enough.

Scarton (2015) presents a project towards document-level QE, by using discourse features. A study on basic discourse features (counts of pronouns, connectives and RST relations) shows high correlation with HTER scores in some scenarios. The problem of finding ideal quality labels for documents is also discussed in this work.

Carpuat and Simard (2012) provide a detailed evaluation of lexical consistency in translations of documents produced by a statistical MT (SMT) system, i.e., on the consistency of words and phrases in the translation of a given source text. SMT was found to be overall consistent in its lexical choices, nearly as consistent as manually translated texts.

Meyer and Webber (2013) present a study on implicit discourse connectives in translation. The phenomenon is evaluated using human references and machine translations for English-French and English-German. They found that humans translated explicit connectives in the source (English) into implicit connectives in the target (German and French) in 18% of the cases. MT systems translated explicit connectives into implicit ones less often.

Li et al. (2014) study connectives in order to improve MT for Chinese-English and Arabic-English. They show that the presence of explicit connectives correlates with high HTER for Chinese-English only. Chinese-English also showed correlation between ambiguous connectives and higher HTER. When comparing the presence of discourse connectives in translations and post-editions, they found that cases of connectives only appearing in the translation or post-edition also show correlation with high HTER scores.

Guzmán et al. (2014) explore RST trees (Mann and Thompson, 1987) for automatic evaluation of MT into English, with a discourse parser to annotate RST trees at sentence level in English. They compare the discourse units of machine translations with those in the references by using tree kernels to compute the number of common subtrees between the two trees. This metric outperformed others at system-level evaluation.

In summary, no previous work has investigated ways to provide a global quality score for an entire document that takes into account document structure, neither for evaluation nor for estimation purposes.

3 Automatic evaluation metrics as quality labels for document-level QE

As discussed in Section 2, although the use of BLEU-style metrics as quality scores for document-level QE clearly seems inadequate, previous work resorted to these automatic metrics because of the lack of better labels. In order to better understand this problem, we conducted an experiment with French-English translations from the LIG corpus (Potet et al., 2012). We took the first part of the corpus containing 119 source documents on the news domain (from various WMT news test sets), their MT by a phrase-based SMT system, a post-edited version of these translations by a human translator, and a reference translation. We used a range of automatic metrics such as BLEU, TER, METEOR-ex (exact match) and METEOR-st (stem match), which are based on a comparison between machine translations and human references, and the "human-targeted" version of BLEU and TER, where machine translations are compared against their post-editions: HBLEU and HTER. Table 1 shows the results of the average score (AVG) for each metric considering all documents, as well as the standard deviation (STDEV).

	AVG	STDEV
BLEU (↑)	0.27	0.05
TER (↓)	0.53	0.07
METEOR-ex (↑)	0.29	0.03
METEOR-st (↑)	0.30	0.03
HTER (↓)	0.21	0.03
HBLEU (↑)	0.64	0.05

Table 1: Average metric scores in the LIG corpus.

We conducted a similar analysis on the English-German (EN-DE) news test set from WMT13 (Bojar et al., 2013), which contains 52 documents, both at document and paragraph levels. Three MT systems were considered in this analysis: **UEDIN** (an SMT system), **PROMT** (a hybrid system) and **RBMT-1** (a rule-based system). Average metric scores are shown in Table 2.

	UEDIN				PROMT				RBMT-1			
	Document		Paragraph		Document		Paragraph		Document		Paragraph	
	AVG	STDEV	AVG	STDEV	AVG	STDEV	AVG	STDEV	AVG	STDEV	AVG	STDEV
BLEU (↑)	0.2	0.048	0.2	0.16	0.19	0.05	0.2	0.16	0.15	0.04	0.16	0.14
TER (↓)	0.62	0.063	0.63	0.24	0.61	0.07	0.62	0.25	0.66	0.06	0.67	0.23
METEOR-ex (↑)	0.37	0.056	0.37	0.16	0.36	0.06	0.37	0.16	0.32	0.05	0.33	0.15
METEOR-st (↑)	0.39	0.058	0.39	0.16	0.38	0.06	0.39	0.16	0.34	0.05	0.35	0.15

Table 2: Average metric scores for automatic metrics in the WMT13 EN-DE corpus.

For all the metrics and corpora, the STDEV values for documents are very small (below 0.1), indicating that all documents are considered similar in terms of quality according to these metrics (the scores are all very close to the mean).

At paragraph level (Table 2), the scores variation increases, with BLEU showing the highest variation. However, the very high STDEV values for BLEU (very close to the actual average score for all documents) is most likely due to the fact that BLEU does not perform well for short segments such as a paragraph due to the n-gram sparsity at this level, as shown in Stanojević and Sima'an (2014).

Overall, it is important to emphasise that BLEU-style metrics were created to evaluate different MT systems based on the same input, as opposed to evaluating different outputs of a single MT system, as

we do here. The experiments in Section 5 attempt to shed some light on alternative ways to accurately measure document-level quality, with an emphasis on designing a label for document-level quality prediction.

4 Experimental settings

As in Scarton et al. (2015), we consider a paragraph as a "document" in the following experiments. This decision was made to make the annotation feasible, given the time and resources available. Although the datasets are different for the two subtasks, they were taken from the same larger corpus and annotated by the the same group of translators.

4.1 Methods

PE1 and PE2 (Section 5) consist in objective assessments through the post-editing of MT sentences in two rounds: in isolation and in context. In the first round (PE1), annotators were asked to post-edit sentences which were shown to them out of context. In the second round (PE2), they were asked to further post-edit the same sentences now given in context and fix any other issues that could only be solved by relying on information beyond individual sentences. For this, each annotator was given as input the output of their PE1, i.e. the sentences they had previously post-edited themselves.

4.2 Data

The datasets were extracted from the test set of the EN-DE WMT13 MT shared task. EN-DE was chosen given the availability of in-house annotators for this language pair. Outputs of the **UEDIN** SMT system were chosen as this was the best participating system for this language pair (Bojar et al., 2013).

For PE1 and PE2, only source (English) paragraphs with 3-8 sentences were selected (filter S-NUMBER) to ensure that there is enough information beyond sentence-level to be evaluated and make the task feasible for the annotators. These paragraphs were further filtered to select those with cohesive devices. Cohesive devices are linguistic units that play a role in establishing cohesion between clauses, sentences or paragraphs (Halliday and Hasan, 1976). Pronouns and discourse connectives are examples of such devices. A list of pronouns and the connectives from Pitler and Nenkova (2009) was considered for that. Finally, paragraphs were ranked according to the number of cohesive devices they contain and the top 200 paragraphs were selected (filter C-DEV). Table 3 shows the statistics of the initial corpus and the resulting selection after each filter.

	Number of Paragraphs	Number of Cohesive devices
FULL CORPUS	1,215	6,488
S-NUMBER	394	3,329
C-DEV	200	2,338

Table 3: WMT13 English source corpus.

For the PE1 experiment, the paragraphs in C-DEV were randomised. Then, sets containing seven paragraphs each were created. For each set, the sentences of its paragraphs were also randomised in order to prevent annotators from having access to wider context when post-editing. The guidelines made it clear to annotators that the sentences they were given were not related, not necessarily part of the same document, and that therefore they should not try to find any relationships among them. For PE2, sentences were put together in their original paragraphs and presented to the annotators as a complete paragraph.

4.3 Annotators

The annotators for both experiments are students of "Translation Studies" courses (TS) in Saarland University, Saarbrücken, Germany. All students were familiar with concepts of MT and with post-editing tools. They were divided in two sets: (i) *Undergraduate students (B.A.)*, who are native speakers

of German; and (ii) *Master students (M.A.)*, the majority of whom are native speakers of German. Non-native speakers have at least seven years of German language studies. B.A. and M.A. students have on average 10 years of English language studies.

PE1 and PE2 were done using three CAT tools: PET (Aziz et al., 2012), Matecat (Federico et al., 2014) and memoQ.[5] These tools operate in very similar ways in terms of their post-editing functionalities, and therefore the use of multiple tools was only meant to make the experiment more interesting for students and did not affect the results.

5 Quality assessment as a two-stage post-editing task

Using HTER, we measured the edit distance between the post-edited versions with and without context. The hypothesis is that differences between the two versions are likely to be corrections that could only be performed with information beyond sentence level.

For PE1, paragraphs from C-DEV set were divided in sets of seven and the sentences were randomised in order to prevent annotators from having access to context when post-editing. For PE2, sentences were put together in their original paragraphs and presented to annotators in context. A total of 112 paragraphs were evaluated in 16 different sets, but only sets where more than two annotators completed the task are presented here (SET1, SET2, SET7, SET9, SET14 and SET15).[6]

5.1 Task agreement

Table 4 shows the agreement for the PE1 and PE2 tasks using Spearman's ρ rank correlation. It was calculated by comparing the HTER values of PE1 against MT and PE2 against PE1. "Annotators" shows the number of annotators per set.

	SET1	SET2	SET5	SET6	SET9	SET10	SET14	SET15	SET16
Annotators	3	3	3	4	4	3	3	3	3
PE1 x MT - HTER	0.63	0.57	0.22	0.32	0.28	0.18	0.30	0.24	0.18
PE1 x PE2 - HTER	0.05	0.07	0.05	0.03	0.10	0.06	0.09	0.07	0.05
PE1 x MT - Spearman	0.52	0.50	0.52	0.56	0.37	0.41	0.71	0.22	0.46
PE2 x PE1 - Spearman	0.38	0.39	−0.03	−0.14	0.25	0.15	0.14	0.18	−0.02

Table 4: HTER values for PE1 against MT and PE1 against PE2 and Spearman's rank correlation values for PE2 against PE1.

The HTER values of PE1 against PE2 are low, as expected, since the changes from PE1 to PE2 are only expected to reflect discourse related issues. In other words, no major changes were expected during the PE2 task. The correlation in HTER between PE1 and MT varies from 0.22 to 0.56, whereas the correlation in HTER between PE1 and PE2 varies between −0.14 and 0.39. The negative figures mean that the annotators strongly disagreed regarding the changes made from PE1 to PE2. This can be related to stylistic choices made by annotators, although further analysis is needed to study that (see Section 5.3).

5.2 Issues beyond sentence level

The values for HTER among annotators in PE2 against PE1 were averaged in order to provide a better visualisation of changes made in the paragraphs from PE1 to PE2. Figure 2 shows the results for individual paragraphs in all sets. The majority of the paragraphs were edited in the second round of post-editions. This clearly indicates that information beyond sentence-level can be helpful to further improve the output of MT systems. Between 0 and 19% of the words have changed from PE1 to PE2 (on average 7% of the words changed).

An example of changes from PE1 to PE2 related to discourse phenomena is shown in Table 5. In this example, two changes are related to the use of information beyond sentence level. The first is related to the substitution of the sentence *"Das ist falsch"* - literal translation of *"This is wrong"* - by *"Das ist*

[5] https://www.memoq.com/

[6] Sets with only two annotators are difficult to interpret.

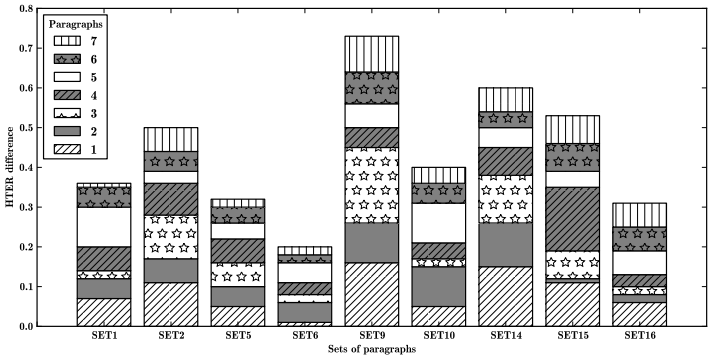


Figure 2: HTER between PE1 and PE2 for each of the seven paragraphs in each set.

PE1: - St. Petersburg bietet nicht viel kulturelles Angebot, Moskau hat viel mehr Kultur, es hat eine Grundlage. Es ist schwer fr die Kunst, sich in unserem Umfeld durchzusetzen. Wir brauchen das kulturelle Fundament, aber wir haben jetzt mehr Schriftsteller als Leser. **Das ist falsch**. In Europa gibt es viele neugierige Menschen, die auf Kunstausstellungen, Konzerte gehen. **Hier ist diese Schicht ist dünn**.
PE2: - St. Petersburg bietet nicht viel kulturelles Angebot, Moskau hat viel mehr Kultur, es hat eine Grundlage. Es ist schwer fr die Kunst, sich in unserem Umfeld durchzusetzen. Wir brauchen das kulturelle Fundament, aber wir haben jetzt mehr Schriftsteller als Leser. **Das ist nicht gut**. In Europa gibt es viele neugierige Menschen, die auf Kunstausstellungen, Konzerte gehen. **Hier ist die Anzahl solcher Menschen gering.**
SRC: - St. Petersburg is not a cultural capital, Moscow has much more culture, there is bedrock there. It's hard for art to grow on our rocks. We need cultural bedrock, but we now have more writers than readers. **This is wrong**. In Europe, there are many curious people, who go to art exhibits, concerts. **Here, this layer is thin.**

Table 5: Example of changes from PE1 to PE2.

nicht gut", which fits better into the context. The other change is related to explicitation of information. The annotator decided to change from *"Hier ist diese Schicht ist dünn"* - literal translation of *"Here, this layer is thin"* - to *"Hier ist die Anzahl solcher Menschen gering"*, a translation that better fits the context of the paragraph *"Here, the number of such people is low"*.

5.3 Manual analysis

In order to better understand the changes made by the annotators from PE1 to PE2 and also better explain the negative values in Table 4, we manually inspected the post-edited data. This analysis was done by senior translators who were not involved in the actual post-editing experiments. They counted modifications performed and categorised them into three classes:

Discourse/context changes: changes related to discourse phenomena, which could only be made by having the entire paragraph text.

Stylistic changes: changes related to translator's stylistic or preferential choices. These changes can be associated with the paragraph context, although they are not strictly necessary under our post-editing guidelines.

Other changes: changes that could have been made without the paragraph context (PE1), but were only performed during PE2.

The results are shown in Table 6. Low agreement in the number of changes and the type of changes among annotators is found in most sets. Although annotators were asked not to make unnecessary

changes (stylistic), some of them made changes of this type (especially annotators 2 and 3 from sets 5 and 6, respectively). These sets are also the ones that show negative values in Table 4. Since stylistic changes do not follow a pattern and are related to the background and preferences of the translator, the high number of this type of change for these sets can be the reason for the negative correlation figures. In the case of SET6, annotator 2 also performed several changes classified as "other changes". This may have also led to negative correlation values. However, the reasons behind the negative values in SET16 could include other phenomena, since overall the variation in the changes performed is low. Further analysis considering the quality of the post-edition needs to be done in order to better explain these results.

	SET1			SET2			SET5			SET6				SET9				SET10			SET14			SET15			SET16		
Annotators	1	2	3	1	2	3	1	2	3	1	2	3	4	1	2	3	4	1	2	3	1	2	3	1	2	3	1	2	3
Discourse/context	2	3	1	0	6	2	2	1	0	2	2	0	0	1	7	1	0	4	0	0	1	0	1	2	1	2	0	1	1
Stylistic	2	0	1	1	0	1	3	11	0	0	3	9	3	5	10	1	3	1	2	2	6	0	0	3	3	2	2	1	3
Other	1	2	4	0	2	2	2	2	6	0	6	0	1	2	0	4	2	1	0	2	2	0	1	1	2	1	1	1	0
Total errors	5	5	6	1	8	5	7	14	6	2	11	9	4	8	17	6	5	6	2	4	9	0	2	6	6	5	3	3	4

Table 6: Manual analysis of PE1 and PE2.

6 Large-scale experiments

In previous sections we have discussed the challenge of assessing document-level quality towards quality prediction. Although the experiment presented in Section 5 showed promising results, the data collected is not suitable for QE approaches. Firstly, there are not enough data points to train a QE model with high confidence in the results. Secondly, the data were annotated by students, which led to several differences in style, since some of them neglected the guidelines. Therefore, in order to use the 2-stage post-editing method, a large-scale data annotation is needed. Moreover, ways to combine the differences between PE1 and PE2 into a quality score should also be explored. In this section, we focus on these two topics, discussing which would be an ideal scenario for training document-level QE models with document-aware quality labels.

Data points In terms of corpora, it is expected numbers around thousands data points. The WMT15 Quality Estimation shared task made available training sets with more than $10,000$ data points for sentence-level and word-level tasks.[7] For document-level, on the other hand, only 800 data points were made available. The reason for this is that it is easier to find more data points for more fine-grained evaluation. Moreover, assessing quality of sentences and words is a more well established task (whilst sentence- and word-level tasks have human targeted scores, the document-level task relies on METEOR). Therefore, there is a lack for both large number of documents to be evaluated and large-scale human targeted document-level assessment.

Regarding the number of documents, we are aware of the difficulty in found enough parallel data with document mark-ups that could be used for QE purposes. Since, traditionally, SMT is done at sentence-level, there is no need for large corpora with large amount of documents (sentences could be randomly placed and only a few documents are need to train a system - the traditional WMT translation shared task relies on data sets with only 52 documents). However, recently, as shown in Section 2, there are more initiatives aiming to improve and evaluate MT outputs at document level. Moreover, the new decoder developed by Hardmeier et al. (2012) works fully at document level, Guzmán et al. (2014) assesses documents at considering information beyond sentences and QuEst has already support for document-level QE. Therefore, there is a lack in the data collection for document-level system development and evaluation. Perhaps, the use of paragraphs (as proposed in this paper) can help in temporally soften he lack of document-level data. The drawback of using paragraphs is that they need to be filtered in order to allow only paragraphs with more than a certain number f sentences, aiming to evaluate phenomena similar in documents.

[7] http://www.statmt.org/wmt15/quality-estimation-task.html

We also intend to use different language pairs for large-large experiments. This would guarantee a more fair evaluation of our method, comparing it across different languages.

Annotation guidelines The task of predicting quality is directly related to which we consider as quality. Previous work, relied on automatic metrics as quality labels, disregarding the problems of such metrics. Therefore, the quality prediction was already biased by the quality of the automatic metrics. Human targeted scores are preferable because they are more reliable and they can assess phenomena that automatic metrics (based in n-gram matches) cannot. Assessing a document as whole is not an easy task, mainly because small problems at word and sentence levels could disturb the judgement of the document quality. Therefore, the proposed 2-stage post-editing method is promising. However, we believe that expert annotators (specialists in post-edition) and a better training could improve the results in terms of annotator's agreement and types of changes.

Evaluation Besides the traditional evaluation of inter annotator agreement, it necessary to find a way to extrinsically evaluate the results of the two-stage post-editing. The aim of this task is to provide information to encoding a quality estimation score at document level. Therefore, the evaluation could be two-folded:

1. **Combination of the difference between PE1 and PE2 with other scores:** one idea is to combine sentence-level HTER (obtained in PE1) with the differences between PE1 and PE2. The problem of this combination is that it cannot be done by using ML techniques (it is not possible to learn a function, since there is no gold standard at document level). Therefore, either we build document-level gold standard scores or we combine the results empirically. Another way to assess document quality could consider a two-stage prediction. Firstly, a sentence-level QE model is built, and the sentences are assessed. Then, these sentence predictions are combined to the PE1 and PE2 difference in order to be used as quality scores for a document-level QE model.

2. **Comparison against traditional metrics:** another way to evaluate the use of the PE1 and PE2 difference is to compare the STDEV of the combined scores with the automatic metrics, aiming to evaluate whether or not the new scores can distinguish documents better than automatic metrics.

Alternative approach Considering a scenario were post-editions in contexts are already available (data obtained from a translation provider, for example), we could consider only apply the first stage of the post-editing method: post-edition of sentences randomly organised (without context). Then, we can measure the difference from the post-edition previous made with the post-edition without context. The same is valid if we find data available with sentences post-edited out-of-context (we could only apply the second stage). However, in order to do this, the guidelines of the previous post-edition and the inter annotator agreement of the previous task should be computed to better understand whether changes made were stylistic or not.

7 Conclusions

This paper focused on judgements of translation quality at document level with the aim to produce labels for QE datasets. We highlighted issues with the use of automatic evaluation metrics for the task, and proposed and experimented with two methods for collecting labels using human annotators.

Our method for collecting labels using human annotators is based on post-editing and showed promising results on uncovering issues that rely on wider context to be identified (and fixed). Although some annotators did not follow the task specification and made unnecessary modifications or did not correct relevant errors at sentence level, overall the results showed that several issues could only be solved with paragraph-wide context. Moreover, even though stylistic changes can be considered unnecessary, some of them could only be made based on wider context.

We will now turn to studying how to use the information reflecting differences between the two rounds of post-editing as labels for QE at document level. One possibility is to use the HTER between the second and first rounds directly, but this can lead to many "0" labels, i.e. no edits made. Other idea is to devise a

function that combines the HTER without context (PE1 x MT) and the difference between PE1 and PE2. PE2 could also be combined with other metrics (e.g. BLEU), in order to define a document-level quality score.

Our findings reveal important discourse dependencies in translation that go beyond QE, with relevance for MT evaluation and MT in general.

Acknowledgements

This work was supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471.

References

Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. Cross-lingual Sentence Compression for Subtitles. In *The 16th Annual Conference of the European Association for Machine Translation*, pages 103–110, Trento, Italy.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *The 43rd Annual Meeting of the Association for Computational Linguistics,*, pages 597–604, Ann Arbor, MI.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *The 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *The Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Marine Carpuat and Michel Simard. 2012. The Trouble with SMT Consistency. In *The Seventh Workshop on Statistical Machine Translation*, pages 442–449, Montreal, Quebec, Canada.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frederic Blain, Philipp Koehn, Christian Buck, and Ulrich Germann. 2014. THE MATECAT TOOL. In *The 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132, Dublin, Ireland.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, Massachusetts.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36:193–202.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using Discourse Structure Improves Machine Translation Evaluation. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, pages 687–698, Baltimore, MD.

Michael A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. English Language Series. Longman, London, UK.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Document-wide Decoding for Phrase-Based Statistical Machine Translation. In *The 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation. In *The 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden.

Junyi Jessy Li, Marine Carpuat, and Ani Nenkova. 2014. Assessing the Discourse Factors that Influence the Quality of Machine Translation. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, pages 283–288, Baltimore, MD.

Chin-Yew Lin and Franz J. Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. In *The 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain.

Willian C. Mann and Sandra A. Thompson. 1987. *Rhetorical Structure Theory: A Theory of Text Organization*. Cambridge University Press, Cambridge, UK.

Thomas Meyer and Bonnie Webber. 2013. Implicitation of Discourse Connectives in (Machine) Translation. In *The Workshop on Discourse in Machine Translation*, pages 19–26, Sofia, Bulgaria.

Karolina Owczarzak, Declan Grove, Josef van Genabith, and Andy Way. 2006. Contextual bitext-derived paraphrases in automatic MT evaluation. In *The NAACL 2006 Workshop on Statistical Machine Translation*, pages 86–93, New York, NY.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *The 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Emily Pitler and Ani Nenkova. 2009. Using Syntax to Disambiguate Explicit Discourse Connectives in Text. In *The Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 13–16, Suntec, Singapore.

Marion Potet, Emmanuelle Esperança-Rodier, Laurent Besacier, and Hervé Blanchon. 2012. Collection of a Large Database of French-English SMT Output Corrections. In *The 8th International Conference on Language Resources and Evaluation*, pages 23–25, Istanbul, Turkey.

Carolina Scarton and Lucia Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *The 17th Annual Conference of the European Association for Machine Translation*, pages 101–108, Dubrovnik, Croatia.

Carolina Scarton, Marcos Zampieri, Mihaela Vela, Josef van Genabith, and Lucia Specia. 2015. Searching for Context: a Study on Document-Level Labels for Translation Quality Estimation. In *The 18th Annual Conference of the European Association for Machine Translation*, pages 121–128, Antalya, Turkey.

Carolina Scarton. 2015. Discourse and document-level information for evaluating language output tasks. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 118–125, Denver, Colorado.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *The Seventh biennial conference of the Association for Machine Translation in the Americas*, AMTA 2006, pages 223–231, Cambridge, MA.

Radu Soricut and Abdessamad Echihabi. 2010. TrustRank: Inducing Trust in Automatic Translations via Ranking. In *The 48th Annual Meeting of the Association for Computational Linguistics*, pages 612–621, Uppsala, Sweden.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *The 13th Annual Conference of the European Association for Machine Translation*, pages 28–37, Barcelona, Spain.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - A translation quality estimation framework. In *The 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL-2013, pages 79–84, Sofia, Bulgaria.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level Translation Quality Prediction with QuEst++. In *The 53rd Annual Meeting of the Association for Computational Linguistics and Seventh International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: System Demonstrations*, Beijing, China.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting Sentence Level Translation Evaluation with Many Dense Features. In *The 2014 Conference on Empirical Methods in Natural Language Processing*, pages 202–206, Doha, Qatar.

Billy T. M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *The 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068. Association for Computational Linguistics, July.

Facilitating SMT with Memory and Dependency Structures

Liangyou Li
School of Computing
Dublin City University, Ireland
liangyouli@computing.dcu.ie

Abstract

Research has suggested that data resources and linguistic knowledge are two important factors that have a strong effect on translation quality. In this paper, we present two methods which improve baseline systems by remembering training data and using dependency structures, respectively. These two methods show a promising dependency-based system with a memory in the future.

1 Introduction

Statistical machine translation (SMT) has been explored for decades. With a strong mathematical foundation, SMT learns models during training and translates new sentences by making a prediction. As well known, SMT largely relies on the data available. The more data given, the better translation quality can SMT achieve. However, corpora compiling is usually very time-consuming. So current research generally focuses on improving SMT by making better use of available data. In this paper, we present two methods which follow this direction in two aspects, respectively: remembering training data and incorporating linguistic knowledge.

Although after trained on given data SMT can be used to translate any given sentence, it is hard to produce a correct translation for a sentence in the training data, because the prediction made by SMT is independent of the data. Such a flaw makes SMT less trusted by translators. Therefore, researchers have been trying to make SMT remember the training data to improve the translation quality (He et al., 2010; Koehn and Senellart, 2010; Ma et al., 2011; Wang et al., 2013). However, previous work on integrating a memory into SMT either treats SMT as a black box or is too complex. In this paper, based on the work of Want et al. (2013), we propose a simple and deep integration of a memory into SMT (Section 2).

Even though a memory can be used to produce high-quality and consistent translations for repetitive materials, the prediction in SMT can cause other problems, such as words-order, agreement and case etc. Thus, researchers have been keeping attention on using linguistic knowledge to solve them, such as using dependency structures (Quirk et al., 2005; Xiong et al., 2007; Xie et al., 2011). Based on the model in Xie et al.(2011), we propose a decomposition method on dependency structures to allow the model to incorporate treelets (Quirk et al., 2005) and non-syntactic phrases (Section 3).

These two methods proposed in this paper do not contradict with each other and thus can be combined in the future. A syntax-based model has a better reordering ability than sequence-based models, including word-based models (Brown et al., 1993) and the phrase-based model (Koehn et al., 2003), we can incorporate a memory into it to improve its translation quality on repetitive materials. This will lead us to a dependency-based model with memory.

2 Remembering Training Data

Example-based machine translation and translation memory have suggested that remembering data and finding similar sentence pairs in the data can be beneficial to translation, especially when the data is repetitive. This motivates us to improve an SMT system by remembering its training data. We call the data as a **TM**.

2.1 Discriminative Framework with TM Features

Generally, in a state-of-the-art statistical translation framework like Moses (Koehn et al., 2007a), the direct translation probability is assigned by a discriminative framework (Och and Ney, 2002). When taking TM into consideration, this framework can be generalized: Equation (1):

$$P(e \mid f, D) = \frac{\exp\{\sum_{m=1}^{M} \lambda_m h_m(e, f, D)\}}{\sum_{e'} \exp\{\sum_{m=1}^{M} \lambda_m h_m(e', f, D)\}}, \tag{1}$$

where D denotes instances (sentence pairs) from TM. Then, we obtain the rule in Equation (2):

$$\begin{aligned} e &= \underset{e'}{\operatorname{argmax}}\{P(e' \mid f, D)\} \\ &\simeq \underset{e'}{\operatorname{argmax}}\{P(e' \mid f, D_f)\} \\ &\simeq \underset{e'}{\operatorname{argmax}}\{\sum_{m=1}^{M} \lambda_m h_m(e', f, D_f)\} \end{aligned} \tag{2}$$

where h_m are feature functions, λ_m are weights.

In this paper, we change features defined in Wang et al. (2013) to TM feature functions and directly add them into a phrase-based system. In decoding, a foreign input sentence f is segmented into a sequence of I phrases $\overline{f}_1^I$, and each foreign phrase $\overline{f}_i$ is translated into a target phrase $\overline{e}_i$. Thus, a TM-related feature function can be seen as the sum of I feature functions which are based on phrase pairs, as in Equation (3):

$$\begin{aligned} h(e, f, D_f) &= h(\overline{e}_1^I, \overline{f}_1^I, D_{\overline{f}_1^I}) \\ &\simeq \sum_{i=1}^{I} h(\overline{e}_i, \overline{f}_i, D_{\overline{f}_1^I}) \end{aligned} \tag{3}$$

where $h(\overline{e}_i, \overline{f}_i, D_{\overline{f}_1^I})$ is measured on the phrase pair $(\overline{e}_i, \overline{f}_i)$ and TM matches $D_{\overline{f}_1^I}$.

2.2 Fuzzy Matching

In this paper, TM-related features are extracted from matches in the TM. For retrieving matches, we use a word-based string edit distance (Koehn and Senellart, 2010) to measure the similarity between an input sentence and a TM instance, as in Equation (4):

$$FMS = 1 - \frac{\text{edi\_distance}(input, tm\_source)}{\max(\mid input \mid, \mid tm\_source \mid)} \tag{4}$$

2.3 TM Features

Wang et al. (2013) propose a deep integration method by using TM information during decoding. They extract features from the best match in the TM and use pre-trained generative models to estimate one or more probabilities and then add them into a phrase-based system for scoring a translation. However, their work requires a rather complex process to obtain training instances for these pre-trained models and needs to define the generative relation between different features. In this paper, we avoid using pre-trained models and tune feature weights to directly maximize BLEU scores (Papineni et al., 2002).

Given an input sentence and its best match in the TM, for each phrase pair applied to the input, we first find its corresponding TM source phrase based on operations for calculating edit-distance. Then, we identify one or more TM target phrases. Then we extract features for the phrase pair. These features are summarized as follows:

- similarity between the best and the input;
- similarity between the source phrase and TM source phrase;

- the length of the source phrase;
- an indicator of whether the source phrase is the punctuation at the end of the input or not;
- similarity between the target phrase and TM target phrases;
- matching and alignment status in context between the source phrase and the TM source phrase;
- alignment status of TM target phrases;
- an indicator of whether a TM target phrase is the longest or not;
- reordering information.

2.4 Multiple Fuzzy Matches

In this paper, besides the best match, we also find a TM instance for each source phrase. We propose a method to find multiple matches to cover as many words in the input as possible: for each source phrase we find a TM instance, which contains this phrase and has the highest fuzzy match score with the input. We call such a TM instance **span-match**. Figure 1 shows an example of finding span-matches. When we extract features for phrase 1, we use TM source 1 and its translation as the match. Similarly, for phrase 2, we use TM source 2; and for phrase 3, we find TM source 3 and use it for feature extraction.

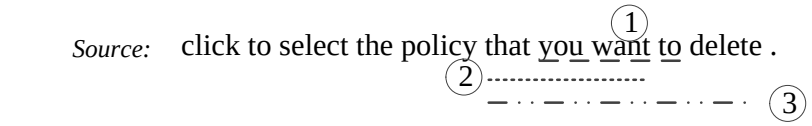


Figure 1: An example of finding multiple matches.

Features from span-matches are similar to those from the best match. We distinguish features from the best match and span-matches. In addition, we also define two more features:

- Feature *NO_SPAN_MATCH* means we cannot find a span-match for the current source phrase.
- Feature *IS_SPAN_BEST* means the span match is equal (the same fuzzy match score) to the best match.

Different to the best match which is estimated over the whole sentence and thus does not bias to any particular source phrase, a span-match provides us information about how a specific source phrase is used and thus may be helpful in selecting a proper target candidate. In addition, note that for a source sentence, the number of span-matches used is flexible, so our method does not need to set a threshold and do optimization on such a parameter.

2.5 Experiment

We conduct experiments on actual TM data as it contains repetitive sentences and is more suitable for testing our method. Our English–Chinese dataset is a translation memory from Symantec with 80K+ training sentences pairs. Our English–French data is from the publicly available JRC-Acquis corpus [1]. On both language pairs, development sets and test sets are randomly selected. Table 1 shows a summary of our data.

[1] http://ipsc.jrc.ec.europa.eu/index.php?id=198

EN-ZH	sentences	words(EN)	words (ZH)
train	86,602	1,148,126	1,171,313
dev	762	10,599	10,791
test	943	16,366	16,375

EN-FR	sentences	words(EN)	words (FR)
train	765,922	20,604,865	22,401,839
dev	1,902	67,403	73,743
test	1,919	71,228	78,177

Table 1: A summary of English–Chinese (EN-ZH) and English–French (EN-FR) corpora

systems	EN–ZH		EN–FR	
	dev	test	dev	test
Phrase-based SMT	52.88	44.63	61.65	61.75
+Wang's model	**54.47**	**45.72**	**62.45**	**62.44**
+TM feature	**54.71**	**45.89**	**62.76**	**62.43**
+multiple fuzzy matches	**55.48***	**46.75***	**63.38***	**63.10***

Table 2: BLEU [%] on English–Chinese (EN-ZH) and English–French (EN-FR) data. Bold figures mean that the result is significantly better than the baseline phrase-based model at $p \leq 0.01$ level. * indicates that multiple fuzzy matches significantly improves the system with TM features at $p \leq 0.01$ level.

We take the phrase-based model in Moses (Koehn et al., 2007b) with default settings as our baseline. Word alignment is performed by GIZA++ (Och and Ney, 2003) with heuristic function *grow-diag-final-and*. We use SRILM (Stolcke, 2002) to train a 5-gram language model on the target side of the training data with modified Kneser-Ney discounting (Chen and Goodman, 1996). Bootstrap resampling (Koehn, 2004) is performed to compute the statistical significance with 1000 iterations. We implement Wang et al. (2013)'s method in Moses for comparison. This method firstly needs to train three models[2] with the factored language model toolkit (Kirchhoff et al., 2007) over a feature sequence of phrase pairs.

Table 2 shows our experiment results on two language pairs. We found that our system with TM features achieves comparable results (+0.24/+0.31 on the dev set and +0.17/-0.01 on the test set) with Wang et al. (2013) and both systems are significantly better than the baseline. After multiple fuzzy matches are incorporated, our system brings further significant improvement (+0.76/+0.62 on dev and +0.86/+0.67 on test).

In addition, we are also interested in the performance of systems on different ranges based on fuzzy match scores. The results are shown in Figure 2. It is easy to see that our system with multiple fuzzy matches achieves the best performance over most ranges. Especially, on the English–Chinese task, when both Wang's model and the TM features are ineffective on the range (0.0,0.4) and [0.4,0.6), multiple fuzzy matches improve the system to give the best translation on both language pairs. However, in the highest range, Wang et al. (2013)'s method gives the best results. It appears that our system does not bias to a high-scoring fuzzy match range and treat all ranges fairly.

3 Decomposing Dependency Structures

Dependency structures have been used in SMT for several years. As it has the best inter-lingual phrasal cohesion properties (Fox, 2002), it is believed to be helpful in translation. For example, Shen et al.

[2]Three probabilities in model III which brings the best performance in their paper:

$$p(TCM \mid SCM, NLN, LTC, SPL, SEP, Z)$$
$$p(LTC \mid CSS, SCM, NLN, SEP, Z)$$
$$p(CPM \mid TCM, SCM, NLN, Z)$$

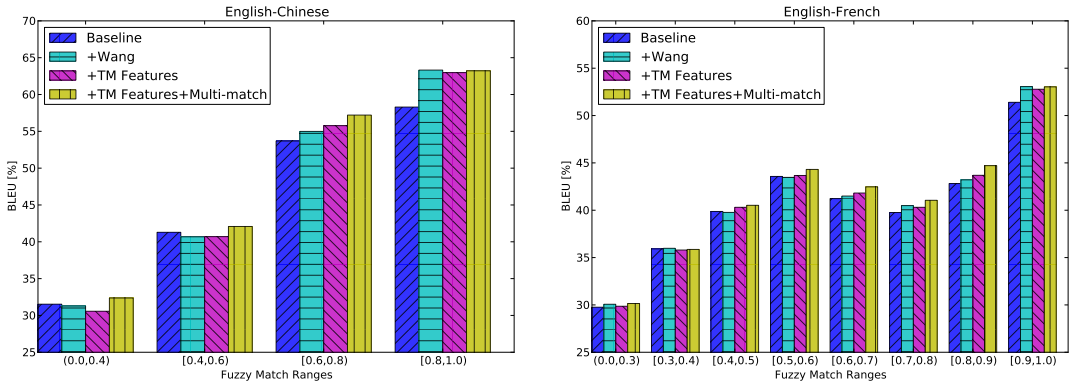


Figure 2: BLEU [%] for different fuzzy match ranges on two language pairs. The baseline is the phrase-based SMT system. The other three systems integrate different TM information into the baseline.

(2010) present a string-to-dependency model by using dependency fragments of neighboring words on the target side, which makes it easier to integrate a dependency language model. However, such string-to-tree systems run slowly(Huang et al., 2006). Menezes and Quirk (2005) and Quirk et al. (2005) propose a treelet (arbitrary connected sub-graph) approach and use dependency structures on the source side. Xiong et al. (2007) extend the treelet approach to allow dependency fragments with gaps. However, these methods need another heuristic or separate reordering model to decide the best target position of the inserted words.

The system used in this section is based on a dependency-to-string (Dep2Str) model (Xie et al., 2011). This model specifies reordering information in its rules and can perform a fast translation. For easily implementing this model, we transform the input dependency tree into a corresponding constituent tree. Then, we enrich this model via decomposing dependency structures. Table 3 shows a glossary for examples used in this section.

Chinese	English
Boliweiya	Bolivia
Juxing	holds
Zongtong	presidential
Yu	and
Guohui	parliament
Xuanju	elections

Table 3: A Chinese-to-English glossary.

3.1 Dependency-to-String Model

In the Dep2Str model, a head-dependent (HD) fragment, which is composed of a head node and all of its dependents, is the basic unit. Two kinds of rules are used. One is the head rule which translates a source word. The other one is the HD rule which consists of three parts: the HD fragment s of the source side, a target string t and a one-to-one mapping from variables in s to variables in t.

Figure 3 shows a derivation for translating a Chinese sentence into an English string in this model. The derivation proceeds from top to bottom. Variables in the higher-level HD rules are substituted by the translations of lower HD fragments recursively.

For easily implementing the Dep2Str model in the popular framework Moses, we transform a dependency tree into a corresponding constituent tree, where source words are leaf nodes and all non-leaf nodes covering a phrase are labeled with categories which are variables defined in a tree-based model.

In the Dep2Str model, each variable represents a word (for the head and leaf node) or a sequence of continuous words (for the internal node). Thus, we use these variables to label non-leaf nodes of the

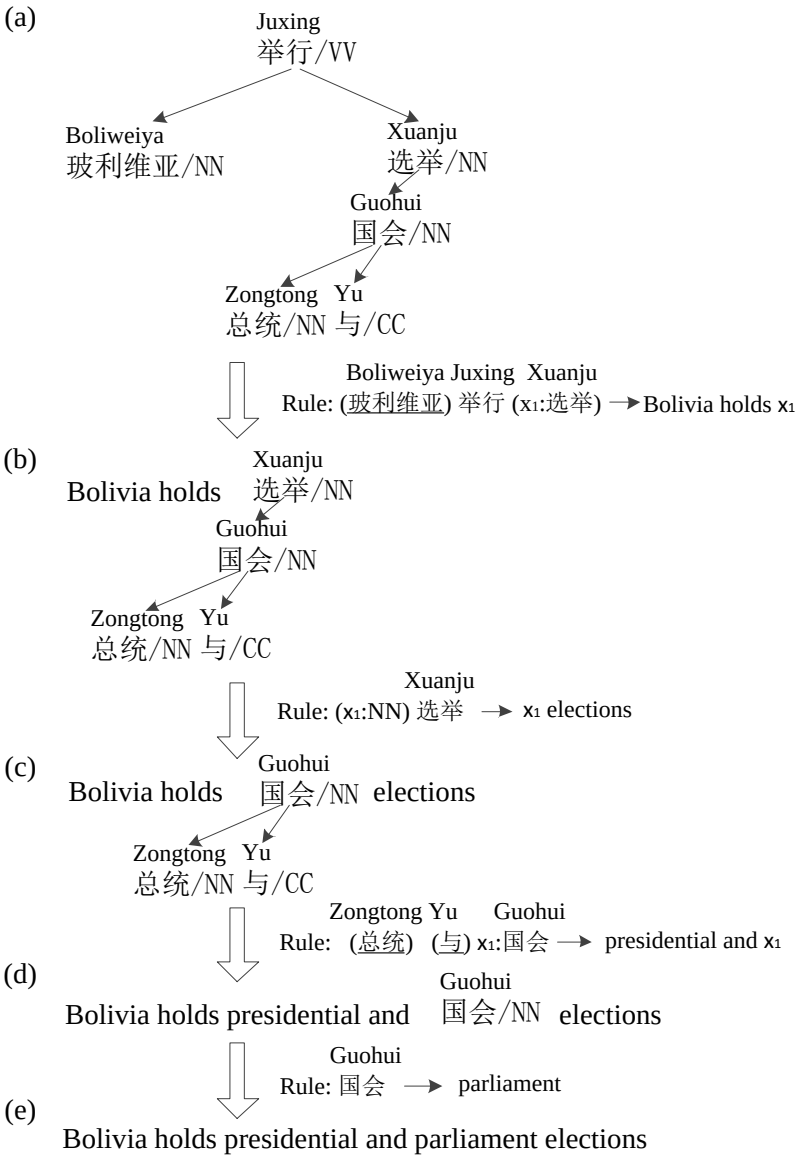


Figure 3: An example of a derivation in Dep2Str which translates a Chinese dependency tree into an English String. Underlined elements indicate leaf nodes.

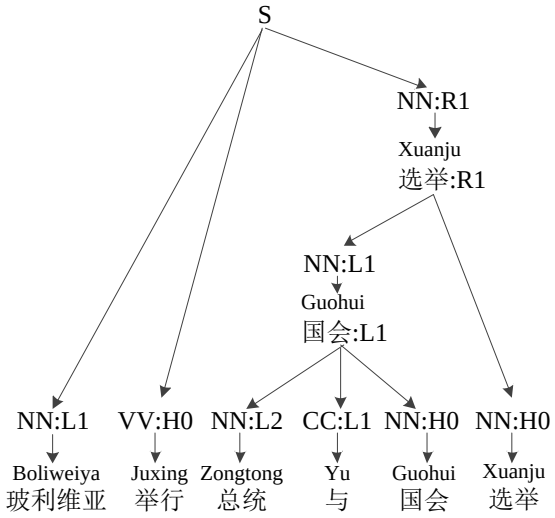


Figure 4: The corresponding constituent tree after transforming the dependency tree in Figure 3.

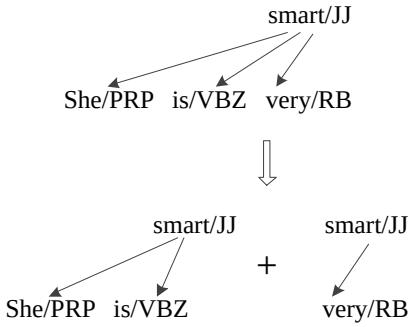


Figure 5: An example of decomposition on a head-dependent fragment.

produced constituent tree. Furthermore, the created nodes are constrained by the dependency information in the HD fragment. Taking the dependency tree in Figure 3 as an example, its transformation result is shown in Figure 4.

3.2 Decomposition of Dependency Structures

The Dep2Str model treats a whole HD fragment as the basic unit, which may result in a sparsity problem. Thus, inspired by the treelet approach (Menezes and Quirk, 2005; Xiong et al., 2007), we define that each HD fragment is decomposed into two smaller parts. This decomposition can be formulated as Equation (5):

$$
\begin{aligned}
& L_i \cdots L_1 H R_1 \cdots R_j \\
& \quad = L_m \cdots L_1 H R_1 \cdots R_n \\
& \quad + L_i \cdots L_{m+1} H R_{n+1} \cdots R_j \\
& \text{subject to} \\
& \quad i \geq 0, j \geq 0 \\
& \quad i \geq m \geq 0, j \geq n \geq 0 \\
& \quad i + j > m + n > 0
\end{aligned}
\tag{5}
$$

where H denotes the head node, L_i denotes the ith left dependent and R_j denotes the jth right dependent. Figure 5 shows an example.

We take advantage of this decomposition in two ways:

- *Sub-structural Rules*

ZH–EN	sentences	words(ZH)	words(EN)
train	1,501,652	38,388,118	44,901,788
dev	878	22,655	26,905
MT04	1,597	43,719	52,705
MT05	1,082	29,880	35,326

DE–EN	sentences	words(DE)	words(EN)
train	2,037,209	52,671,991	55,023,999
dev	3,003	72,661	74,753
test12	3,003	72,603	72,988
test13	3,000	63,412	64,810

Table 4: Chinese–English (ZH–EN) and Germa–English (DE–EN) corpora.

We extract sub-structural rules by taking each possible sub-fragment as a new HD fragment, which are used directly in the model.

- *Pseudo-Forest*

  For an HD fragment in the input dependency tree, we can translate one of its sub-fragments first, then obtain the whole translation by combining with translations of another sub-fragment, as shown in Figure 6. We encode the decomposition into the input dependency tree which results in a pseudo-forest. Figure 7 shows an example.

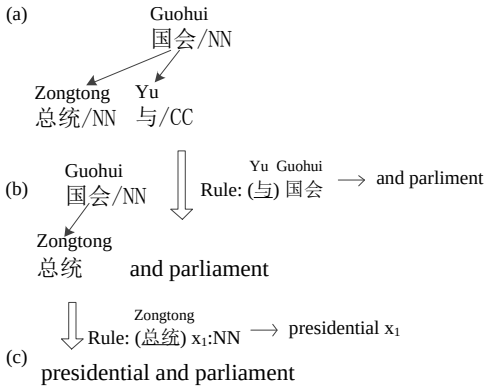


Figure 6: An example of translating a large HD fragment with the help of translations of its decomposed fragments.

3.3 Experiments

Our Chinese–English training corpus has 1.5M+ sentence pairs from the LDC data, including LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, the Hansards portion of LDC2004T08 and LDC2005T06. We take NIST 2002 as the development set to tune weights, and NIST 2004 (MT04) and NIST 2005 (MT05) as the test data to evaluate the systems. Our German–English training corpus is from WMT 2014, including Europarl V7 and News Commentary. News-test 2011 is taken as the development set, while News-test 2012 (test12) and News-test 2013 (test13) are our test sets. Table 4 shows a summary of our data.

In the Chinese–English translation task, the Stanford Chinese word segmenter (Chang et al., 2008) is used to segment Chinese sentences into words. The Stanford dependency parser (Chang et al., 2009) parses a Chinese sentence into the projective dependency tree. We tokenize German sentences with scripts in Moses and use mate-tools[3] to perform morphological analysis and parse the sentence (Bohnet,

[3] http://code.google.com/p/mate-tools/

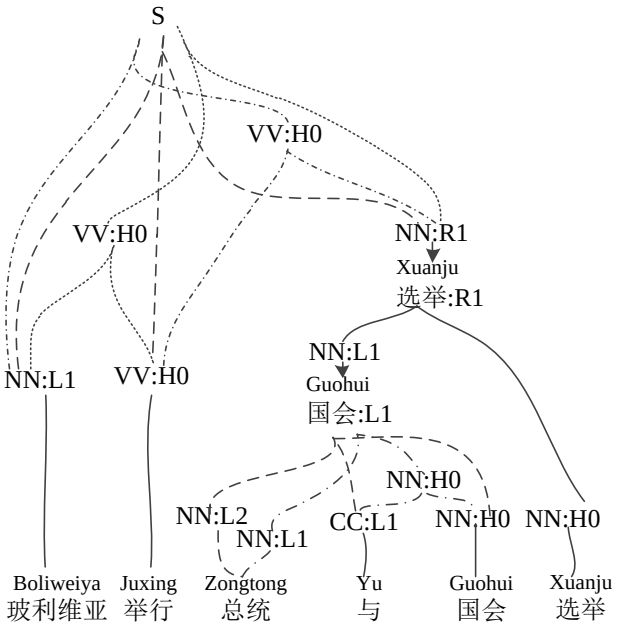


Figure 7: An example of a pseudo-forest. Edges drawn in the same type of line are owned by the same sub-tree. Solid lines are shared edges.

Systems	ZH–EN		DE–EN	
	MT04	MT05	test12	test13
Moses HPB	35.56	33.99	20.44	22.77
D2S	33.93	32.56	20.05	22.13
+pseudo-forest	**34.28**	**34.10**	19.98	21.68
+sub-structural rules	**34.78**	**33.63**	**20.52**	**22.76**
+phrase	–	–	**20.91***	**23.46***
+pseudo-forest	**35.46**	**34.13**	20.25	22.24
+phrase	**36.76***	**34.67***	**20.75***	**23.20***

Table 5: BLEU score [%] of our methods and Moses HPB on Chinese–English (ZH–EN) and German–English (DE–EN) tasks. We use bold font to indicate that the result of our method is significantly better than D2S at $p \leq 0.01$ level, and * to indicate the result is significantly better than Moses HPB at $p \leq 0.01$ level.

2010). Then the MaltParser[4] converts the parse results into projective dependency trees (Nivre and Nilsson, 2005).

Word alignment is performed by GIZA++ with the heuristic function *grow-diag-final-and*. We use SRILM to train a 5-gram language model on the Xinhua portion of the English Gigaword corpus 5th edition with modified Kneser-Ney discounting. Bootstrap resampling is performed to compute the statistical significance with 1000 iterations.

Table 5 shows the translation results. On the Chinese–English task, we find that the decomposition approach, including sub-structural rules and pseudo-forest, improves the baseline system D2S significantly (absolute improvement of +1.53/+1.57). As a result, our system achieves comparable (-0.1/+0.14) results with hierarchical phrase-based model (Chiang, 2005) in Moses. After including phrasal rules, our system performs significantly better (absolute improvement of +1.2/+0.68) than Moses HPB on both test sets.

On the German–English task, Table 5 shows that incorporating sub-structural rules improves the baseline D2S system significantly (absolute improvement of +0.47/+0.63), and achieves a slightly better

[4] http://www.maltparser.org/

Systems	# Rules	
	CE task	DE task
Moses HPB	388M	684M
D2S	27M	41M
+sub-structural rules	116M	121M
+phrase	215M	274M

Table 6: The number of rules in different systems On the Chinese–English (CE) and German–English (DE) corpus. Note that pseudo-forest (not listed) does not influence the number of rules.

(+0.08) result on test12 than Moses HPB. However, the pseudo-forest produces a negative effect on the baseline system (-0.07/-0.45), despite the fact that our system combining both methods together is still better (+0.2/+0.11) than the baseline D2S. In the end, by resorting to phrasal rules, our system achieves the best performance which is significantly better (absolute improvement of +0.47/+0.59) than Moses HPB.

Besides long-distance reordering (Xie et al., 2011), another advantage of the Dep2Str model is its simplicity. It can perform fast translation with fewer rules than HPB. Table 6 shows the number of rules in each system. It is clear that all of our systems use fewer rules than HPB. However, the number of rules is not proportional to translation quality, as shown in Table 5.

4 Conclusion

In this paper, we present methods for incorporating a memory and linguistic knowledge in SMT. For remembering training data, we present a discriminative framework which can integrate the data into SMT. In this framework, we add more feature functions, which model relations between a source sentence and matched instances in the memory, into a phrase-based SMT. In experiments on English–Chinese and English–French tasks, our method performs significantly better than the baseline phrase-based system. Furthermore, we present a method to efficiently use multiple fuzzy matches. Experiments show that this addition significantly improves our system.

For a better use of dependency structures, we propose to decompose large structures into smaller pieces. Based on a dependency-to-string model (Dep2Str), this decomposition enriches the model with more rules during training and allows us to create a pseudo-forest as an input instead of a dependency tree during decoding. Large-scale experiments on Chinese–English and German–English tasks show that this decomposition can significantly improve the baseline Dep2Str model on both language pairs. The code is now available on `http://computing.dcu.ie/~liangyouli/dep2str.zip`.

These two methods lead us to a future promising model which integrates memory with a dependency-based model. The resulting system would have a better reordering ability provided by dependency structures and can effectively translate repetitive materials by using the memory.

Acknowledgements

Liangyou Li is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471. Thanks to Qun Liu, Andy Way and Jun Xie for their help on this work.

References

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative Reordering with Chinese Grammatical Relations Features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59, Boulder, Colorado.

Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Santa Cruz, California.

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan.

Heidi J. Fox. 2002. Phrasal Cohesion and Statistical Machine Translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, pages 304–3111, Philadelphia.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden, July.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A Syntax-directed Translator with Extended Domain of Locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8, New York City, New York.

Katrin Kirchhoff, Jeff Bilmes, and Kevin Duh. 2007. Factored Language Models Tutorial. In *UWEE Technical Report*, Department of Electrical Engineering, University of Washington, June.

Philipp Koehn and Jean Senellart. 2010. Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver, Colorado, USA, November.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007a. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007b. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Prague, Czech Republic.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.

Yanjun Ma, Yifan He, Andy Way, and Josef van Genabith. 2011. Consistent Translation using Discriminative Learning - A Translation Memory-Inspired Approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1239–1248, Portland, Oregon, USA, June.

Arul Menezes and Chris Quirk. 2005. Dependency Treelet Translation: The Convergence of Statistical and Example-Based Machine-translation? In *Proceedings of the Workshop on Example-based Machine Translation at MT Summit X*, September.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106, Ann Arbor, Michigan.

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Philadelphia, PA, USA.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-Dependency Statistical Machine Translation. *Computational Linguistics*, 36(4):649–671, December.

Andreas Stolcke. 2002. SRILM-an Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 257–286, Denver, Colorado, USA, November.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2013. Integrating Translation Memory into Phrase-Based Machine Translation during Decoding. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Sofia, Bulgaria, August.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A Novel Dependency-to-string Model for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 216–226, Edinburgh, United Kingdom.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2007. A Dependency Treelet String Correspondence Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 40–47, Prague, June.

A Component-Centric Design Framework for Translation Interfaces

Chris Hokamp
Dublin City University, Ireland
chokamp@computing.dcu.ie

Abstract

We propose a framework for designing and testing CAT tools, called *Component-Centric Design and Optimization*, proposing a factorization of translation interfaces into components that can be tuned and optimized individually. Current Computer-Aided Translation (CAT) interfaces lack a principled design paradigm which provides a framework for new components to be consistently tested, compared, and optimized based on user feedback — the proposed framework provides a useful functional abstraction over the components that comprise a translation interface. We also present several novel translation components which integrate existing NLP technologies into localisation workflows. Our new components include prototypes leveraging linked data resources, and services inspired by SMT systems which are guided by user input.

1 Introduction

The purpose of a translation interface is to facilitate the mapping of sequences in a source language into sequences in a target language. The ways in which the source to target mapping can occur, and the optimal means for humans to interact with and modify translation data models are open research topics. In this work, we consider translation within the context of Human-Computer Interaction. This approach focuses upon the design of tools with translators integrated as critical components of the system (O'Brien, 2012) — the goal of the interface is to provide the translator with interactions which facilitate the mapping of source language sequences into target language sequences.

The ability to view a source segment and to compose and persist (save) a corresponding target segment are the baseline functionalities required for a translation interface. Most existing CAT tools provide a set of standard interactions, such as editing of target segments, confirming segments, and searching a glossary, within a segment-oriented view. The additional features provided by a CAT tool are either focused upon providing translators with metadata to facilitate the translation process, or upon assisting users in other stages of the localisation workflow, such as document conversion, handling markup and tags, or terminology extraction.

For the purpose of this paper, we define *Translation Resources* as functions which transform text sequences in one language to text sequences in another language. Translation Resources are the primary class of *data services* used by translation interfaces (see below). Machine Translation (MT) and Translation Memory (TM) technologies are examples of Translation Resources that are commonly integrated into localisation processes. All modern CAT tools provide users with graphical interfaces to MT and TM resources. These resources may be remote services accessed by web APIs, or they may be integrated into the translation interface, with all data storage and computation performed locally. The *target* representation resulting from applying a Translation Resource to a source segment may be directly utilized as a translation if it is deemed to be of sufficient quality, or it may be further transformed by a human expert with knowledge of both the source and the target languages, or by a monolingual domain expert, which is referred to as *post-editing*.

The rest of this paper is organized as follows: Section 2 introduces the Component-Centric view of translation interfaces. Section 3 introduces HandyCAT, our implementation of a modular CAT tool. Section 4 discusses localisation standards, and introduces our novel components for linked

data integration and source language preauthoring. Section 5 presents a user study of autocomplete components for English-Spanish translation. Section 6 presents conclusions, and describes our plans for future work.

2 A Component-Centric View of CAT Interfaces

The purpose of any component within a CAT tool is to facilitate modification of the state of the translation data model. The translation data model is hierarchical, object-oriented view of the translation data, which can be serialized into XML, JSON, or another structured representation. Translation interfaces maintain a mutable internal representation of the data as the user translates, which can be mapped into a standard format such as XLIFF once the translation job is finished. User modification of the data model is accomplished via interactions such as text input or selection from a list of translation options.

The impact of a component integrated into a CAT tool or a localization pipeline must be empirically measurable with respect to a convincing metric. In the case of CAT tools, the most common and obvious metric is translation speed, but even this is not trivial to measure, because of differences in translation difficulty between each segment, and because of individual differences between translators.

In the context of Computer Aided Translation, we experiment with several different editing scenarios: (1) Translation from scratch, where translators are expected to compose the full target segment, with or without aid from additional components such as autocompletion engines, translation memories and glossaries, (2) post-editing, where translators are expected to alter the output of a machine translation system until it is perceived to be human-quality text, and (3) source side pre-authoring, where a translator or monolingual content creator composes source text using a constrained language, with the intent to produce better-quality machine translations in the target language. For each of these editing scenarios, we design specialized components designed to make tedious or time-consuming parts of the translation task more efficient.

We define a *Component* as a means of transforming and/or rendering some data to the end user, optionally with interactive capabilities which allow users to modify the underlying data model. We factor translation interfaces into two primary component types: *Data Services* which are services that transform input data into another representation, and *Interaction Elements*, which provide the means for users to view and possibly modify parts of the data model. A component may be composed of other sub components, and typically consists of both Interaction Elements and Data Services.

An example of a data service is an SMT component which takes a source sentence and outputs one or more target hypotheses; an example of an interaction component is a text editing area designed to aid the user in post-editing an SMT hypothesis.

Because this work is primarily focused upon enhanced tooling for CAT workflows, we further decompose the interaction component types into *display components* and *input components*. A display component's only job is to render data to the user (a one way interaction), whereas an input component both renders data, and allows the user to change the underlying datamodel via one or more interactions.

A complete translation interface is composed of data services and interaction components, with the minimal orchestration layer between components. This abstract, component-oriented view of the interface allows dynamic configuration based on user needs, and enables each component to be analyzed and enhanced individually, simplifying some of the daunting complexity of the interface.

2.1 A Formalization Of Component-Centric Optimization

An Interface is a set of (E+, D*) tuples, where:
E = Interaction Element
D = Data Service
+ = one or more
* = zero or more

Interaction Elements and Data Services are parametrized by vectors of scalar values such that the complete configuration of the interface consists of the set of parameter vectors which specify all of

the components in the interface. Figure 1 shows an example object tree for the area of the translation interface which provides interaction at the segment level of the data model.

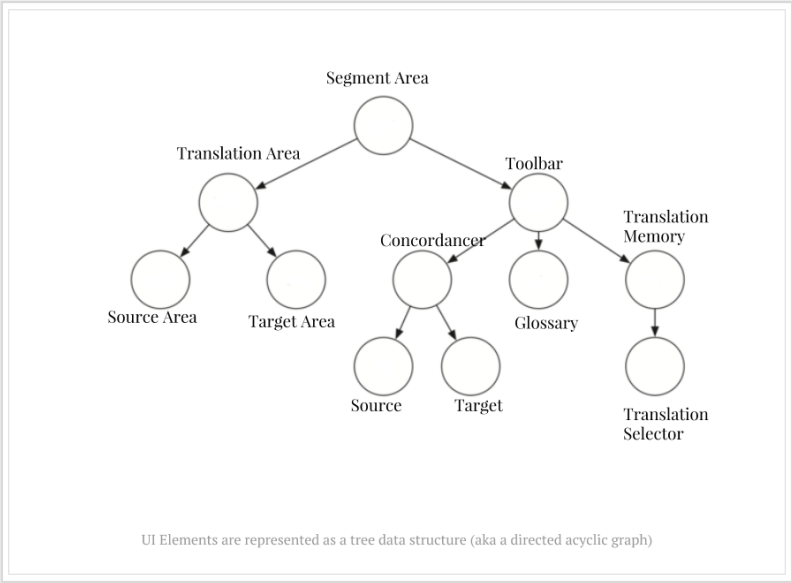


Figure 1: A component factorization for the segment area of a CAT interface

Component-oriented design allows a principled approach to CAT tool development. Instead of viewing an interface as a static tool, we view the CAT tool as a composition of nested components, which are mostly standalone. The only job of the container elements is to act as 'glue' between components, and to render the area depending on the display capabilities of the user's device.

The Component-oriented view allows each element or collection of elements to be optimized separately, drastically lowering the complexity of the optimization process.

2.2 Translation Resources

Data services that can map from source to target segments are *Translation Resources*, regardless of the method of mapping they employ, whether it involves decoding (searching) to find an optimal target sequence with respect to some scoring function, such as in a log-linear discriminative framework for SMT (Koehn, 2010), or matching based upon string distance, such as edit distance (Navarro, 1999), as used in translation memory and concordance components.

Although SMT and TM are the translation resources in widespread use today, in fact translation resources are on a continnuum of matching functions which span the range from exact matches to pure generation via real-valued distributed representations (Cho et al., 2014).

In general, translation technologies can be combined in an *internal*, or an *external* manner. Internal combination means that technologies are integrated within a single module, for example as components of a log-linear model which scores hypotheses in an SMT decoder (Koehn, 2010). External combination allows the constituent modules to remain distinct, but adds a layer which coordinates the technologies, producing integrated output for downstream use. The optimal means of coordination and integration depends upon the downstream usecase of the translations produced by the component.

In practice, most translation resources have a dynamic component which tries to account for unseen tokens or sub-sequences in the source sequence, in order to expand the coverage of the sequences in the dataset. Even the fuzzy matching logic used by a Translation Memory is a means of increasing the recall of the component at the expense of precision. The implicit hypothesis motivating translation resources which support partial matching is that post-editing a small portion of a target translation candidate will be easier than translating the source segment from scratch. However, the ideal threshold for fuzzy matching depends on many factors, and is generally manually tuned to the usecase.

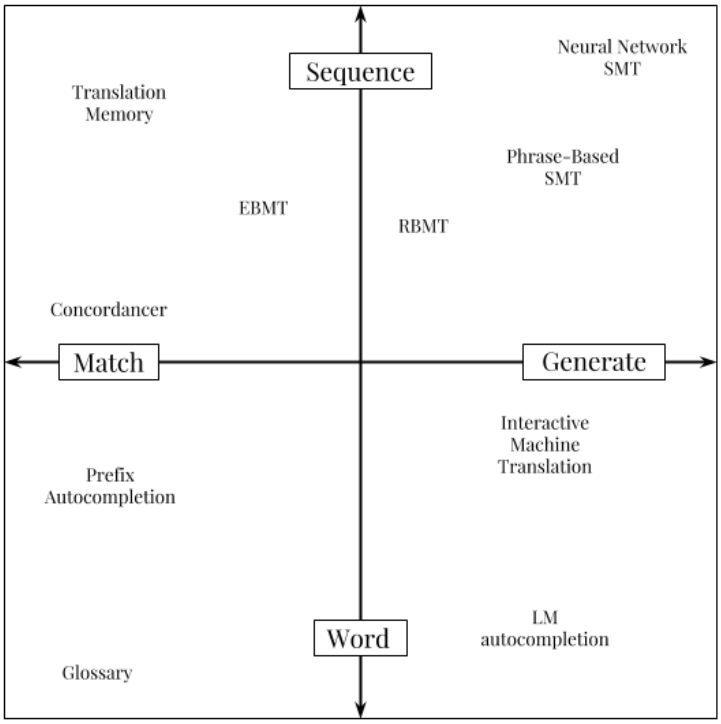


Figure 2: The Translation Resource continuum —x-axis is the output generation method, y-axis is output length

We propose a simple visualization of Translation Resources in a two-dimensional space, where one axis measures the degree of string matching vs. generation via statistical models, and the other axis measures the length of the generated sequence, from single words to complete sequences. Figure 2 maps some common Translation Resources into this space.

2.3 Integrating Translation Technologies

The services that are composed into translation resources can be integrated within a single framework, which orchestrates all components, or each component can be implemented as a standalone service. The service-oriented view has many advantages over a single monolithic system, especially as components become more complex. The primary design consideration for integrated translation components is the downstream use of the data. For instance, is the output of the translation system intended for use as is, or will it be presented to translators for post-editing? Is the translation system running in batch mode, where some delay between input and output is acceptable, or does the system need to operate in near-real time? In the case where the system must output translations in real-time, some sacrifices in performance may be necessary in order to obtain the desired response time.

Modern Computer Aided Translation (CAT) tools make use of many backend services, such as translation memories, glossaries, machine translation systems, and text analytics tools. These services may run locally, on the same system as the user interface, or they may be remote services which are accessed via web APIs.

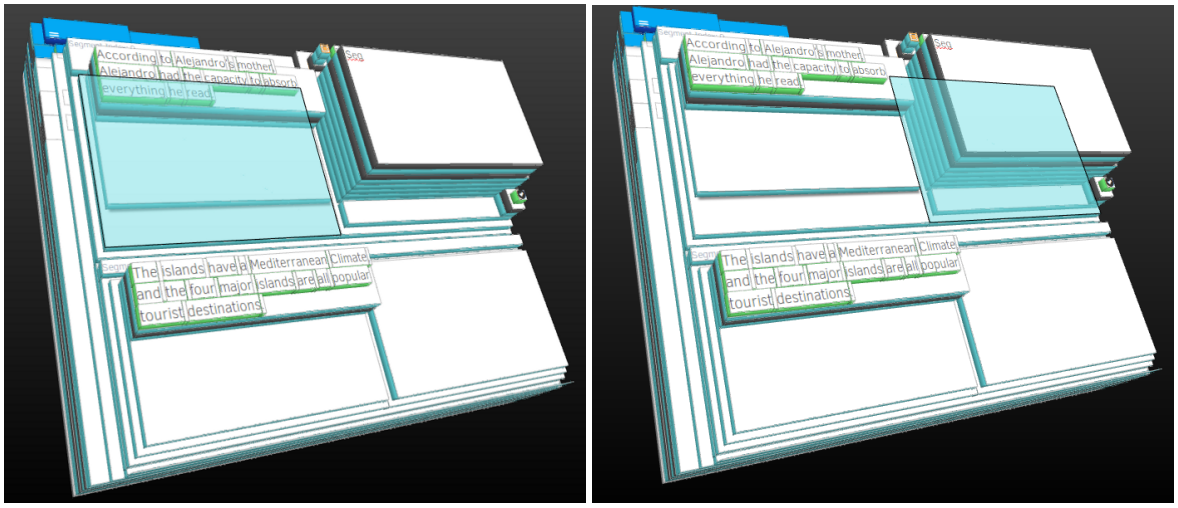
3 HandyCAT — A Modular CAT Tool

HandyCAT is our implementation of a flexible web based CAT tool (Lewis et al. 2014), specifically designed with interoperability and extensibility in mind. Current CAT tools are designed as self-contained platforms which attempt to implement all important features within a monolithic framework. Although many existing tools provide a full suite of features which satisfy all core user needs, this design methodology is not conducive to research into new components, because the implementation phase requires significant modification of the platform source code.

With HandyCAT, new graphical elements and data services can easily be added and removed from the interface, and components can directly plug into the relevant part of the translation data model. Thus, it is an ideal platform for developing prototypes and conducting user studies with new components.

The core functions implemented by a CAT tool include displaying the the source segment, and allowing the user to edit a target sentence, either by typing from scratch, or by post-editing a TM segment or an MT hypothesis. By abstracting over the common graphical affordances (Greeno, 1994) leveraged by CAT tools, we identify a set of *Functional Areas*, which provide an abstract representation of the graphical areas that render translation data to the user. Any translation interface will include areas dedicated to these functions, but the interactions implemented by the components within the functional areas may be drastically different from implementation to implementation.

The core set of containers common to all CAT tools with a two-dimensional graphical interface includes the source area, the target area, and the tooling area. Note that parameters which control the graphical rendering of the areas, i.e. their position and size on the page, are hyperparameters of the application, and can be determined by tuning/optimization, trial and error, or constraints imposed by the user's device. On devices with smaller form factors, such as mobile devices, it is likely that most areas will be hidden by default, and accessed via specific interactions, whereas devices with more display area may display all areas simultaneously.



(a) Source Area

(b) Target Area

Figure 3: HandyCAT container areas

HandyCAT provides several predefined functional areas which have access to a specific part of the data model or document tree. Developers can specify which parts of the datamodel their component can access by placing it inside a container, such as the *EditArea*, or the *TargetArea*. By defining the component at the right level in the hierarchy, developers can ensure that their components are isolated, and that the document model is always synchronized with UI state. Figure 3 visualizes the container hierarchy in HandyCAT.

3.1 Synchronized UI state and XLIFF datamodel

HandyCAT's internal data model is inspired by the XLIFF 2.0 standard, and can be mapped directly to and from XLIFF 1.2 or 2.0. Each element in the XML DOM is represented as an object — components explicitly specify their interfaces, i.e. which types of objects they can interact with.

Tooling components also have access to interface state in addition to the document model, so they can respond to segments being activated or deactivated, or external events.

Each component can also specify its own keyboard shortcuts and logging formats, as well as any remote services it must access. A new translation interface can be composed of the components which are needed for an experiment, with customizable logging.

Component Interfaces
Document
Segment
Source
Target
Global Tooling
Segment Tooling

Table 1: Examples of component interfaces

4 Integrating NLP Technologies into CAT Workflows

4.1 Localisation and Linked Data Standards

Modern localisation workflows present significant software development challenges (Lewis et al., 2009). Over the last decade, several standards have emerged which have the potential to make data exchange between different stages of the localisation workflow more efficient, scalable, and reliable. These include XLIFF 1.2[1] and 2.0[2], the Internationalization Tagset (ITS)[3], and the NLP Interchange Format (NIF)[4].

The XLIFF 2.0 standard defines a Document Object Model (DOM) which provides an object-oriented interface to translation documents. The HandyCAT datamodel is isomorphic to the core elements of the XLIFF standard, and component interfaces are directly mappable to elements in the XLIFF DOM. This design provides a clear component hierarchy, and enables a modular approach to component development and testing.

The standardisation of localisation interchange formats is vital to enable the integration of translation technology into complex systems. The end goal of standardization is to achieve *Content Interoperability* across content producers. (Lewis et al., 2014) evaluates current and emerging standards for interoperability, introducing a pipeline for localisation workflows with interchange between components facilitated through localisation standards such as XLIFF 1.2 and 2.0, and the Internationalization Tagset (ITS) version 2.0.

4.2 Using Linked Data for Dynamic Terminologies

Resources such as DBpedia and Freebase (Bollacker et al., 2008) are examples of open knowledge bases which take advantage of the implicit and explicit links in Wikipedia and other resources to construct a graph of entities with edges encoding relationships between the entities.

Linked data resources are promising ways to map terminology between languages in a consistent and scalable manner (Hokamp, 2015). Linked Open Data (LOD) can potentially be utilized at many points in the localization workflow. By augmenting the metadata for the source or target text in a pre- or post-processing phase, linked data can provide metadata which facilitates human translation and quality assessment. Where metadata can be added in a completely automatic way, the task of determining whether or not the data is useful in the context can be pushed to the translator, who can decide where and how to make use of the additional information. The feedback from translators can then be used to augment the knowledge base.

We present a prototype application of linked data within a CAT component, called a *Linked Dynamic Terminology*. This approach leverages statistical named entity extraction and surface form labeling across languages to create a terminology which uses source language context to rank terms. Figure 4 shows the stages in the dynamic terminology workflow.

The dynamic terminology component combines a LOD resource and a statistical entity linker within a translator-in-the-loop system. Translator-in-the-loop means that the design of the system explicitly

[1] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html
[2] http://docs.oasis-open.org/xliff/xliff-core/v2.0/os/xliff-core-v2.0-os.html
[3] http://www.w3.org/TR/its20/
[4] http://persistence.uni-leipzig.org/nlp2rdf/

includes a human, who is finally responsible for selecting the correct translation. This setup can be contrasted with a fully automatic design, where the target sentence would automatically be augmented with terminology, either via a machine translation system, or via an automatic post-editing phase.

The dynamic linked terminology component is designed as a standalone module that can easily added or removed from HandyCAT. The server components are designed as microservices which are accessed using RESTful APIs, each fulfilling a single task in the dynamic terminology building process. The system is designed to operate in realtime, meaning that it does not require any offline preprocessing of the translation job.

The process of finding the target-language surface form for a source entity requires two disambiguation steps. The first step is *entity linking*, where the entity extraction system attempts to link surface forms in the source language to the specific entity they represents. See (Daiber et al., 2013) for details on the algorithm used to determine which entity is most likely represented given a surface form and a surrounding context.

The second step is *entity labeling*, where the translator selects the correct surface form for the entity in the target language. This requires retrieving the set of target language links for each entity, and making them available in the translation interface.

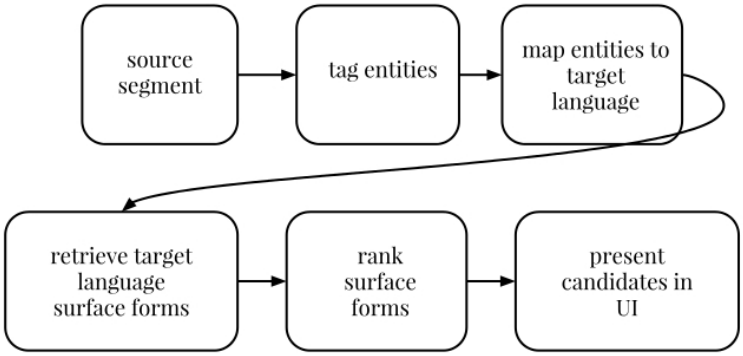


Figure 4: The dynamic linked terminology workflow

In our design, the linking system tagger detects entities in a source segment, and the LOD resource provides candidate translations in the target language. By leveraging Wikipedias multilingual graph through the DBPedia datasets, the system can provide suggestions for many language pairs. The multilingual graph of entities is thus transformed into a dynamic terminology database. Figure 5 shows a screenshot from an actual editing session using the linked terminology component.

The term dynamic in this context means that the set of suggestions for a term depend upon the context in which it is being used. Because the disambiguation is done with respect to the source context, the possible target forms are ranked according to their likelihood with respect to the underlying entity. A central hypothesis of this work is that this dynamic re-ranking provides a major improvement over the standard glossary or terminology lookup, which can only look for string matches for a particular token or phrase, without regard to the particular sense of the term in context.

4.3 ProphetMT — Syntactic MT-driven Preauthoring

Preauthoring is an approach to content authoring which aims to produce source content that is easy to translate into one or more target languages. We have developed ProphetMT, a monolingual authoring tool which allows users to easily compose an in-domain sentence with the help of tree-based SMT-driven auto-suggestions. The interface also visualizes target language sentences as they are built by the SMT system. When the user is finished composing, the final translation(s) are generated by a tree-based SMT sytem using the text and structural information provided by the user. With this domain-specific controlled language, ProphetMT will produce highly reliable translations. The contributions of this work are: (1)

Figure 5: Linked Terminology Screenshot

we develop a user friendly auto-completion based editor which guarantees the vocabulary and grammar chosen by user are compatible with a tree-based SMT model, and (2) by applying a shift-reduce like parsing feature, this editor allows users to write from left to right and generates the parsing results on the fly. Therefore with this in domain composing restriction as well as the parsing result, a highly reliable translation can be generated.

Although current machine translation methods have improved rapidly in the past decade, SMT is still not reliable enough to be considered human-quality without significant post-editing (OBrien, 2005). The primary reason is that natural languages are full of ambiguities.

We refer to the task of source-language composition as "Computer Aided Authoring" (CAA). This name was chosen to reflect the association with Computer Aided Translation, in that the goal of the interface is to guide the user in creating a source text with additional structural metadata which will make machine translation of the text easier.

Post-editing requires bilingual experts who need much more training than monolingual writers. Combining the composition and translation components of the content creation pipeline makes the authoring process much more efficient and cost-effective. Therefore, computer-aided monolingual authoring tools are a promising way to alleviate some of the unnecessary labour in post-editing.

All existing computer-aided authoring tools within a translation context employ a kind of interactive paradigm with a controlled language (CL). Mitamura (1999) allow users to compose from scratch, and discuss the issues in designing a CL for rule based machine translation. (Power et al., 2003) describes a CL authoring tool for multilingual generation. (Marti et al., 2010) is a rule based rewriting tool which does syntactic analysis. (Mirkin et al., 2013) introduces a confidence-driven rewriting tool which is inspired by (Callison-Burch et al., 2006; Du et al., 2010) that paraphrases the OOVs or the "hard-to-translate-part" of the source side in order to improve SMT performance.

CL can be seen as a subset of natural language which is designed for writing clear technical documentation in a particular domain (Power et al., 2003). The advantages of applying CL are straightforward: clear and consistent composition guidelines as well as less ambiguity in translation. However, the problems are also obvious: designing the rules usually requires human linguists, and rules may be difficult for end-users to grasp. The sentences that can be generated are often limited in length and complexity. Finally, the expressiveness of the writer is also greatly constrained, which may or may not be a problem depending upon the domain.

Preauthoring can be especially useful in constrained language domains, where the usage of certain

phrase structures or terminology in the source language will make target translations much more accurate.

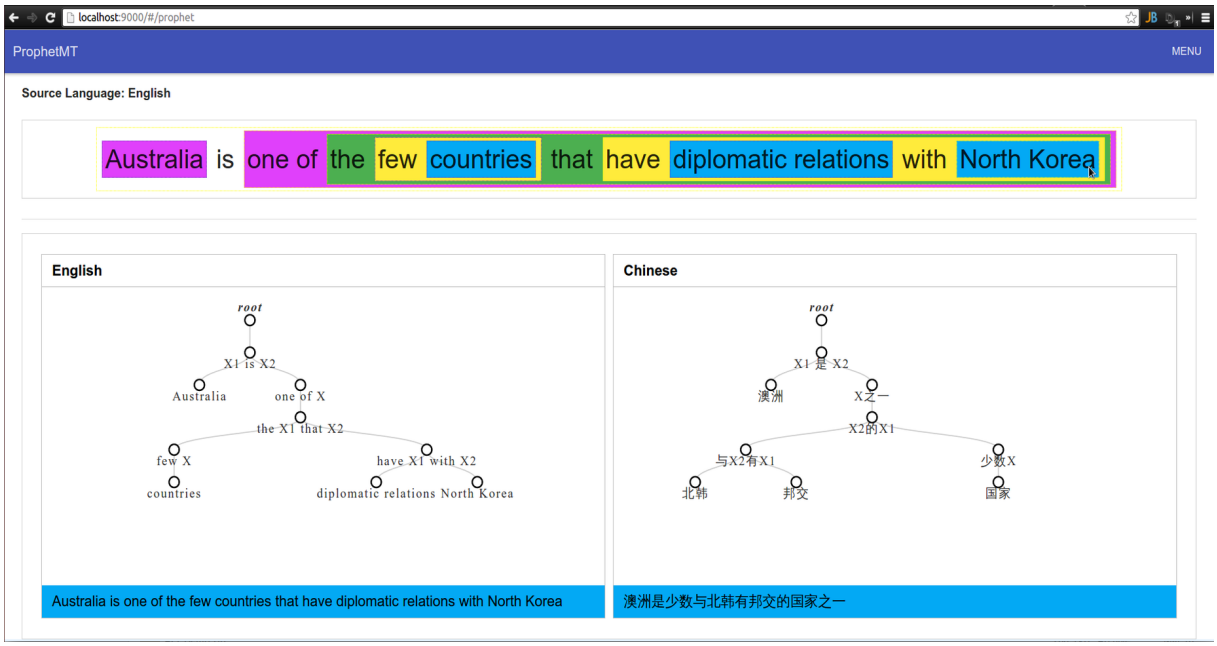


Figure 6: A screenshot from the ProphetMT preauthoring interface

Figure 6 shows a screenshot from the ProphetMT interface.

5 Optimizing Autocompletion and Typeahead Components

Autocompletion has become an indispensable part of modern text processing tools, especially those designed for use with mobile devices. Despite the ubiquity of autocompletion in content creation workflows, there is very little research evaluating autocompletion within the context of Computer-Aided Translation. There are two major benefits to autocompletion when compared with composition from scratch: (1) autocompletion may help the user to find better ways to translate the source segment by providing a variety of suggestions, and (2) autocompletion may save the user time by minimizing the number of input operations needed to complete a translation. In the following experiments, we focus on (2), attempting to design a system which helps users to complete a translation task more quickly.

This work presents a simple but effective approach to autocompletion or "type-ahead" components based on elements of a traditional statistical machine translation system. We compare two autocompletion engines in a user study with 16 translation students who are native Spanish speakers. The *baseline autocompleter* uses prefix matching over the known vocabulary of the target language. This mimics the autocompletion utilities typically found in smartphones or word-processing software. The *phrase table backed autocompleter* leverages the SMT phrase table and a target-side language model to provide enhanced suggestions to translators. When a translator enters a segment, the possible completions are first retrieved from a data service which queries a Moses phrase table. As the user translates, the system dynamically reranks the completion candidates using a language model conditioned on the current target prefix.

When compared to autocompletion strategies using Interactive Machine Translation (Green et al., 2014; Bender et al., 2005), this approach has the advantage that it can be implemented without deploying a full-fledged machine translation system. Phrase table autocompletion systems are also much easier to scale than IMT systems.

Although IMT arguably provides better suggestions by directly leveraging all features implemented in the SMT decoder, current IMT systems are difficult to scale beyond a few users, because the computational requirements of the decoder are very demanding. Stack based SMT decoders (cite Koehn 2009) generally have a tradeoff between computational requirements and quality, which can be controlled

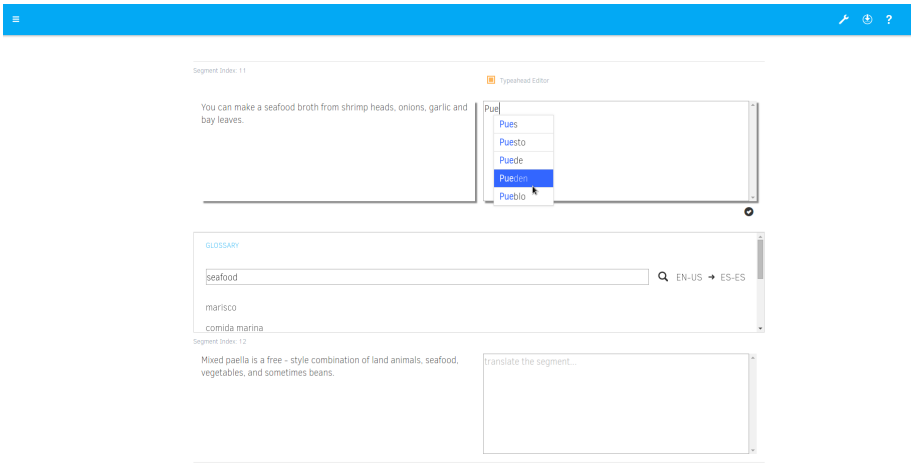


Figure 7: A target-side autocomplete component backed by a phrase table and language model.

by the hyperparameters of the system, which, for stack-based decoders include the stack size and the maximum n-gram size of the language model.

Our system effectively uses the language model as the only feature in the decoder, and allows the user to quickly filter the phrase options by typing the prefix of the desired word or phrase. Despite this simplification, we observe a significant improvement in translation speed when compared to a baseline prefix autocompletion engine.

5.1 Experiment Design

The baseline for our experiments is a monolingual autocomplete which has no knowledge of the source segment being translated. Although this engine is naive to the source, it can arguably still make translators substantially faster, by saving them keystrokes.

Dataset	Avg. Time
A	79.53
B	76.47

Table 2: Average sentence completion time for each dataset

Table 2 shows that the average completion times for each sentence in both datasets were quite similar. We interpret this to mean that the difficulty of the sentences in the two datasets were comparable.

Autocomplete Type	Avg. Time
Default	82.75
PT-backed	73.25

Table 3: Average sentence completion time for each autocomplete type

We discard the worst outliers for each test segment, in order to account for possible mistakes during the translation process, such as late confirmation of a segment.

We selected 30 sentences from the English Wikipedia, which were divided into two datasets of 15 sentences each. Because there are two possible orderings of the translation tasks, and two autocomple utilities to test, we created eight experimental groups which account for every permutation of task ordering and autocomplete configuration. Participants were randomly assigned to one of the eight groups.

Table 3 shows the average completion time for a segment using the two autocomplete types. The PT-backed autocomplete allows translators to complete a segment more than 9 seconds faster on average.

According to the our experimental results, translators are more than 10% faster with the enhanced

autocomplete. This is an encouraging finding, especially given the relative simplicity of the phrase-table and LM-backed autocompletion components.

6 Conclusions And Future Work

Our work to date has enabled us to propose a flexible and scalable design methodology for building Computer Aided Translation tools, called *Component-Centric Design and Optimization*. Component Centric Optimization is a principled method for improving user interfaces based on implicit feedback from user interactions. By viewing CAT tools as combinations of standalone components, the problem of optimization becomes more tractable, and components designed with this approach are automatically reusable within other interfaces with minimal modification.

We have also implemented prototypes of novel components making use of NLP technologies to enhance user efficiency and to improve various aspects of CAT tool usability. Several user studies have been conducted in order to evaluate some of our new components. The enhanced autocompletion component using an SMT phrase table and language model enabled a significant improvement in translation speed in a realistic usage scenario.

In order for any data service to be useful in a real CAT workflow, it must be able to function in near real-time, otherwise it will not be useful to translators, and its applications to dynamically generated content will be very limited. Despite this critical requirement, most machine translation research attempting to improve performance with respect to well established metrics such as BLEU (Papineni et al., 2002) or METEOR (Lavie and Denkowski, 2009) is still performed in a batch-processing scenario with predetermined test data, and state-of-the-art results can currently only be achieved with systems that require vast computational resources.

Research systems are also typically not optimized for speed and scalability — critical requirements for including translation technologies into user-facing systems. The integration of translation systems into real-time frameworks is likely to require significant redesign, and implementations generally must make tradeoffs between response time and output quality, where the optimal setting is dependent upon the end use case. User-facing components must also be designed to be robust against unexpected formats or data types, issues which are traditionally outside the domain of MT research.

Future work will focus upon formalizing the optimization process for CAT components, and upon designing and testing new components for specific translation tasks. We are currently extending our autocompletion experiments to include a state-of-the-art Interactive Machine Translation system. We also plan to develop and test several new graphical components. Finally, we will continue to improve ranking algorithms for providing in-context autocompletion capabilities.

Acknowledgements

Chris Hokamp is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013). ProphetMT is a collaboration with Xiaofeng Wu. Many thanks to Qun Liu for helpful supervision and guidance, and to Josef Van Genabith and Kashif Shah for their very useful reviews. We also thank Sharon O'Brien and Joss Moorkens for their feedback on early versions of HandyCAT.

References

Oliver Bender, David Vilar, Richard Zens, and Hermann Ney. 2005. Comparison of generation strategies for interactive machine translation. In *In Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation*, pages 30–40.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA. ACM.

Jinhua Du, Jie Jiang, and Andy Way. 2010. Facilitating translation using source language paraphrase lattices. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 420–429. Association for Computational Linguistics.

Spence Green, I. Sida Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and D. Christopher Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236. Association for Computational Linguistics.

James G. Greeno. 1994. Gibson's affordances. *Psychological Review*, pages 336–342.

Chris Hokamp. 2015. Leveraging nlp technologies and linked open data to create better cat tools. *Localisation Focus - The International Journal of Localisation*, 14.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115, September.

David Lewis, Stephen Curran, Kevin Feeney, Zohar Etzioni, John Keeney, Andy Way, and Reinhard Schäler. 2009. Web service integration for next generation localisation. In *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '09, pages 47–55, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Lewis, Qun Liu, Leroy Finn, Chris Hokamp, Felix Sasaki, and David Filip. 2014. Open, web-based internationalization and localization tools. *Translation Spaces*, 3(1):99–132.

J.M. Marti, D. Ahs, B.K. Lee, J.R. Falkena, J.A. Nelson, B. Kohlmeier, F. Liger, R. Pamarthi, C.B. Lerum, V. Mody, et al. 2010. User interface for machine aided authoring and translation, May 4. US Patent 7,711,546.

Shachar Mirkin, Sriram Venkatapathy, Marc Dymetman, and Ioan Calapodescu. 2013. Sort: An interactive source-rewriting tool for improved translation. In *ACL (Conference System Demonstrations)*, pages 85–90. Citeseer.

Teruko Mitamura. 1999. Controlled language for multilingual machine translation. In *Proceedings of Machine Translation Summit VII, Singapore*, pages 46–52.

Gonzalo Navarro. 1999. A guided tour to approximate string matching. *ACM Computing Surveys*, 33:2001.

Sharon O'Brien. 2012. Translation as human-computer interaction. *Translation Spaces*, 1(1):101–122.

Sharon OBrien. 2005. Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation*, 19(1):37–58.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Power, Donia Scott, and Anthony Hartley. 2003. Multilingual generation of controlled languages.

On Using Syntactic Preordering Models to Delimit Morphosyntactic Search Space

Joachim Daiber
Institute for Logic, Language and Computation (ILLC)
University of Amsterdam, The Netherlands
j.daiber@uva.nl

Abstract

Source-side reordering (or preordering) approaches have recently seen a surge in popularity in machine translation research. The advantages of this approach lie in the often enormous reductions in translation time and in empirically good results in translation quality in various language pairs. For many language pairs, however—especially for translation into morphologically rich languages—the assumptions of such models may be too crude. On the other hand, the more complex models that such language pairs call for might increase the search space to an extent that would diminish their advantages. In this paper, we examine the question whether it is feasible to use the aforementioned purely syntactic preordering models as a means to delimit the search space for more complex morphosyntactic models. In order to do so, we propose a preordering model based on a popular preordering algorithm (Lerner and Petrov, 2013). This new preordering model is able to produce both *n*-best preorderings as well as distributions over possible preorderings in the form of a lattice and is therefore a good fit for use by subsequent morphosyntactic models.

1 Introduction

In recent years, a significant amount of research in machine translation has focused on methods for effectively restricting the prohibitively large search space of phrase-based statistical machine translation systems. One popular method providing a crude but theoretically motivated restriction of this space is preordering (also pre-reordering or source-reordering). In preordering, the source sentence is rearranged to reflect the assumed word order in the target language. This provides an effective method of handling word and phrase movements caused by long-range dependencies, which usually enlarge the search space significantly. After preordering, decoding can be performed in fully monotone or close to monotone fashion, making the method applicable to a wide range of translation systems, including ngram-based translation (Marino et al., 2006) and recent approaches to neural machine translation (Bahdanau et al., 2015, inter alia). While systems using this approach have in the past not always been able to show improvements in translation quality over systems using more exhaustive search algorithms or specialized reordering models, preordering provides several benefits: Apart from facilitating the integration of additional information sources such as paraphrases, preordering approaches provide significant improvements in runtime performance. Jehl et al. (2014), for example, report an 80-fold speed improvement using their preordering system compared to a standard system producing translations of the same quality.

preordering systems can be compared along several dimensions. The main distinctions are whether the reordering rules are specified manually (Collins et al., 2005) or automatically learnt from data (Lerner and Petrov, 2013; Khalilov and Sima'an, 2012). Furthermore, approaches differ in the types of syntactic structures they assume. Systems may use only source or target syntax (Lerner and Petrov, 2013; Khalilov and Sima'an, 2012), both source and target syntax or no syntax at all (e.g. DeNero and Uszkoreit (2011)).

In this paper, we focus on approaches using only source-side syntax. Following Lerner and Petrov (2013), we use source-side dependency trees. Dependency grammar offers a flexible and light-weight syntactic framework that can cover a large number of languages and provides suitable syntactic representations for reordering.

The annotation conventions of the training treebank and hence the form of the dependency trees produced by the parser play a significant role for the preordering system; hence, we will briefly describe the treebank format in Section 4.1. In Section 2, we review related work. Section 3 introduce the model we propose for delimiting the preordering space and the general framework, including the integration of non-local features. Section 4 presents results of the experimental evaluation and a discussion of these results. We conclude in Section 5.

2 Related work

Various approaches to preordering have been explored in the literature. We will give a brief overview of work establishing the background for the presented method and then focus on approaches with and without source syntax.

2.1 Gold experiments

In order to investigate the upper bounds of preordering in terms of quality and integration with translation systems, several researchers have performed studies with gold reorderings. Khalilov and Sima'an (2012), as well as Herrmann et al. (2013) compare various systems and provide oracle scores for syntax-based preordering models. These studies show that given *perfect gold* reorderings estimated via automatic alignment of the test set enables the translation systems enormous jumps in translation quality and further provides improvements in the model size of downstream translation models. Additionally, it was found that properties of the source syntax representation, e.g. how deeply phrase structure trees are nested, can significantly hamper the quality of the preordering.

2.2 Preordering with source syntax

Jehl et al. (2014) learn order decisions for sibling nodes of the source-side parse tree and explore the space of possible permutations using a depth-first branch-and-bound search. In later work, this model is further improved by replacing the standard logistic regression with a feed-forward neural network (de Gispert et al., 2015). This modification shows both improved empirical results and eliminates the need for feature engineering. Similarly, Lerner and Petrov (2013) learn classifiers to permute the tree nodes of a dependency tree. The main difference here is that the permutation of up to 6 tree nodes is predicted directly instead of the orientation of individual node pairs. Figure 1 shows an example dependency tree that can serve as input to the system.

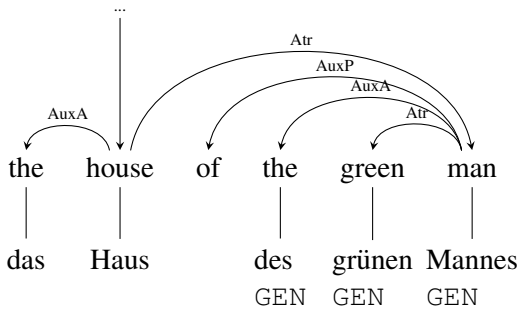


Figure 1: Translation of English PP as genitive NP in German.

2.3 Preordering without syntax

Tromble and Eisner (2009) apply machine learning techniques to learn ITG-like orientations (straight or inverted order) for each pair of input words in the sentence. The best reordering is then determined using an $O(n^3)$ algorithm based on chart parsing.

Generally, systems not relying on syntactic information range from simple approaches such as the application of multiple MT systems in which a MT system learns the preordering (i.e. one MT system for

$\mathbf{s} \rightarrow \mathbf{s}'$, and one for $\mathbf{s}' \rightarrow \mathbf{t}$, Costa-jussà and Fonollosa (2006)) to more advanced systems automatically inducing parse trees from the aligned data (DeNero and Uszkoreit, 2011).

3 Generating the preordering space

Our work is closely oriented on the work of Lerner and Petrov (2013), in which feature-rich classifiers are trained to directly predict the word order based on the source-side dependency parse tree. This is done by traversing the dependency tree in a top-down fashion and predicting the target order for each tree family (a family consists of a syntactic head and its dependents/children). To address sparsity issues, two models are introduced. For each subtree, the 1-step model directly predicts the target order of the child nodes. Unlike other preordering models, which often restrict the space of possible permutations, e.g. by the permutations permissible under the ITG constraints (Wu, 1997), the space of possible permutations for each sub-tree is restricted to the k permutations most commonly observed in the data. The blowup in permutation space with growing numbers of children is addressed by a second model, the 2-step model. This model decreases the number of nodes involved in any single word order decision. A binary classifier (the pivot classifier) first predicts whether a child node should occur to the left or to the right of the head of the subtree. The order of the set of nodes to the left and to the right of the head is then directly predicted as in the 1-step model.

3.1 Preordering beyond first-best predictions

The cascade-of-classifiers approach exhibits the problematic characteristic that classification errors occurring near the top of the tree will propagate disproportionately to later decisions. The goal of this work is to be able to pass the decision to a more complex morphosyntactic model. Hence, this issue will become problematic. In order to address this problem, we extract *n*-best preorderings from the classifier decisions. A distribution over the *n*-best preordered sentences can then be passed to the subsequent model or directly to a machine translation decoder either as a list of options or in the form of a lattice. Similar to the practice of *n*-best list extraction in MT decoders such as Moses, the preordering problem likewise allows the extraction of *n*-best preordering options either with or without additional integration of a language model.

Given a source sentence $\mathbf{s}$ and a corresponding dependency parse tree τ, π denotes a permutation of the tree nodes. We define the score of a preordering $\mathbf{s}'$ as follows

$$P(\mathbf{s}' \mid \mathbf{s}, \tau) = \prod_{h \in \tau} P_T(\pi_n \mid \mathbf{s}, h, \tau) \tag{1}$$

where

$$P_T(\pi \mid \mathbf{s}, h, \tau) = P(\psi \mid \mathbf{s}, h, \tau) \times P_L(\pi_L \mid \mathbf{s}, h, \tau) \times P_R(\pi_R \mid \mathbf{s}, h, \tau) \tag{2}$$

For each dependency tree family, the generative story of this model is as follows: First, decide on the positions of the child nodes relative to the head, i.e. $P(\psi \mid \mathbf{s}, h, \tau)$. Then, decide the order of the nodes on the left, $P_L(\pi_L \mid \mathbf{s}, h, \tau)$, and on the right, $P_R(\pi_R \mid \mathbf{s}, h, \tau)$.

The following modifications to the preordering algorithm are performed: For each family with head h in the source-side dependency tree τ, we extract the best k_T local preorderings using the function PREORDERFAMILY in Algorithm 1. $\Psi(cs)$ is the set of possible choices when distributing nodes using the pivot classifier. Given a set of child nodes cs, $\Pi(cs)$ is the set of possible permutations for all nodes. The best permutations for the left and right side are extracted by the following methods:

$$\pi_L \leftarrow \underset{\pi_L \in \Pi(cs_L)}{\arg \text{bestk}} \; P_L(\pi_L \mid \mathbf{s}, h, \tau) \tag{3}$$

$$\pi_R \leftarrow \underset{\pi_R \in \Pi(cs_R)}{\arg \text{bestk}} \; P_R(\pi_R \mid \mathbf{s}, h, \tau) \tag{4}$$

Note that since this model is implemented using multi-class classifiers, finding the best k_O permutations for the nodes to the left and right of the head, i.e. Equation 3 and 4, only require one multi-class classification.

Algorithm 1 N-best preordering of a source-tree family

function PREORDERFAMILY(h, τ)
  $cs \leftarrow$ CHILDREN(h, τ)
  $topk \leftarrow$ PRIORITYQUEUE()

  for $\psi \leftarrow \underset{\psi \in \Psi(cs)}{\arg\,\text{bestk}}\ P(\psi(cs) \mid \mathbf{s}, h, \tau)$ **do**
    $cs_L \leftarrow$ LEFT(ψ)
    $cs_R \leftarrow$ RIGHT(ψ)
    for $\pi_L \leftarrow \underset{\pi_L \in \Pi(cs_L)}{\arg\,\text{bestk}}\ P_L(\pi_L \mid \mathbf{s}, h, \tau)$ **do**
      for $\pi_R \leftarrow \underset{\pi_R \in \Pi(cs_R)}{\arg\,\text{bestk}}\ P_R(\pi_R \mid \mathbf{s}, h, \tau)$ **do**
        $p \leftarrow$ PERMUTATION(ψ, π_L, π_R)
        TOPK.PUSH($P(\psi \mid \mathbf{s}, h, \tau) \times P_L(\pi_L \mid \mathbf{s}, h, \tau) \times P_R(\pi_R \mid \mathbf{s}, h, \tau), p$)
      end for
    end for
  end for
  return TOPK.TAKE(k_T)
end function

Accordingly, Equation 5 is implemented as k-best sequence extraction from a conditional random field classifier.

$$\underset{\psi \in \Psi(cs)}{\arg\,\text{bestk}}\ P(\psi(cs) \mid \mathbf{s}, h, \tau) \tag{5}$$

For each of the maximally k_P possible ways to distribute the child nodes when taking the pivot decision, 2 classifications have to be performed: one for the nodes on the left and one for the nodes on the right. The extraction of *n*-best preorderings therefore requires $2 \times k_P$ classifications for each source-side tree family. The best k_T local permutations for each source-tree family enable *n*-best extraction for the whole tree.

3.2 Integration of non-local features

In a pilot study on English–German translation, we found the independence assumptions of the model to be too strong. The generative process assumes that the preordering occurs only within the constituents defined by the dependency tree. This implies that the words that have been moved to the right and left boundaries of two consecutive tree constituents are not taken into account when making an ordering decision. Previous work on preordering (Khalilov and Sima'an, 2012) has shown that the integration of even a weak trigram language model estimated over the gold preorderings $\mathbf{s}'$ can improve preordering performance.

Since we use projective dependency trees, which are internally converted to a flat phrase structure representation, the model can be expressed in the form of a weighted context-free grammar in which labels encode the order of the constituents. One method to weaken the independence assumptions of this grammar is the direct integration of a language model (LM). This is reminiscent of the integration of the finite state language model with the synchronous context-free grammar used in hierarchical phrase-based translation (Chiang, 2007).

Hence, instead of searching for

$$\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau) \tag{6}$$

the search will now include the ngram language model, such that:

$$\hat{s} = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau) P_{LM}(\mathbf{s}') \tag{7}$$

This integration can be performed in three ways: the simplest form of integration, which is fast but allows for significant search errors, is to generate an *n*-best list of preorderings using the $-$LM (i.e. without the LM or other non-local features) preordering model and re-score this list using the language model. On the other end of the spectrum, the language model can be integrated by performing a full intersection between the preordering CFG and the finite state automaton that defines the language model (Bar-Hillel et al., 1961). While this would allow for exact search, this method is often found to be too slow in practice. A compromise between these two extremes is cube pruning (Chiang, 2007), in which the inner LM cost as well as the left and right LM states are stored on each node, so that it is possible to perform bottom-up dynamic programming to efficiently determine the total LM cost by combining the intermediate node costs.

Keeping the properties required for performing cube pruning, we use the more general log-linear model formulation (Och and Ney, 2002):

$$\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau)^{\lambda_{RM}} P_{LM}(\mathbf{s})^{\lambda_{LM}} ... \tag{8}$$

$$= \arg\max_{\mathbf{s}'} \prod_i \phi_i(\mathbf{s}')^{\lambda_i} \tag{9}$$

$$= \arg\max_{\mathbf{s}'} \sum_i \lambda_i \log \phi_i(\mathbf{s}') \tag{10}$$

As feature functions, we initially use the node preodering score $P(\mathbf{s}' \mid \mathbf{s}, \tau)$, a generic ngram language model over the gold preorderings $\mathbf{s}'$, a language model over part-of-speech tags and a class-based language model.

On every source tree node, cube pruning is performed with a beam size of $k_{+\text{LM}}$ node configurations. The best $k_{-\text{LM}}$ preordering labels are considered for expansion. Additionally, we prune all preordering labels for which the language model cost is higher than the language model cost of the original source tree order. To make individual configurations comparable, we follow Chiang (2007) in adding a heuristic cost that approximates the cost of the first $m-1$ words: $\log P_{\text{LM}}(e_1 ... e_l)$ where $l = \min\{m-1, |e|\}$ for an m-gram language model. In our case, e is the vector of reordered source-side words at a specific tree node. We add the heuristic cost of all relevant feature functions ϕ_i for the set of language model feature functions Φ_{LM} as $\sum_{i \in \Phi_{\text{LM}}} \lambda_i \log \phi_i(e_1 ... e_l)$.

3.3 Additional features

In addition to the basic preordering model and the language model, the formulation as a log-linear model allows the addition of arbitrary feature functions. Here, we describe the additional feature templates added to the model.

Phrase dependency grammar features

A common conception in work within machine translation is that using trees can be problematic because it prescribes a segmentation of a sentence that might not be optimal across languages. It has often been shown that modeling larger units beyond constituent borders provides more beneficial results. Gimpel and Smith (2011) introduce the notion of quasi-synchronous phrase dependency grammars. While previous work has used the notion of phrase dependency trees, the phrases were obtained in a monolingual manner, such as via syntactic chunking (Wu et al., 2009). Gimpel and Smith (2011) define phrase dependency trees over the space of possible phrase segmentations of a standard phrase-based machine translation system. This choice entails that it is not possible to construct a single phrase dependency tree during training, but it allows the definition of the set of phrase dependency trees that are consistent with both the phrase segmentation and a lexical dependency parse tree.

The goal of phrase dependency tree features is to boost preorderings which respect segmentations present in the alignments. Let γ be a segmentation of $\mathbf{s}$ into phrases such that $\forall i, \gamma_i = \langle \mathbf{s}_j ... \mathbf{s}_k \rangle$ and $\gamma_1 ... \gamma_n = \mathbf{s}$. Given $\mathbf{s}$, we first produce the set $\Gamma_{\mathbf{s}}$ of possible phrase segmentations.

Next, we introduce a feature function ϕ_{PhDp}:

$$\phi_{\text{PhDp}}(\mathbf{s}') = \sum_{\gamma \in \Gamma'_{\mathbf{s}}} q(\gamma, \mathbf{s}') \tag{11}$$

where $q(\gamma, \mathbf{s}')$ is a non-negative score for each phrase dependency tree.

$$q(\gamma, \mathbf{s}') = \begin{cases} 1.0 & \text{if } \mathbf{s}' \text{ consistent with } \gamma \\ 0.0 & \text{otherwise} \end{cases} \tag{12}$$

The set of possible phrase segmentations $\Gamma_{\mathbf{s}}$ of the input sentence is determined by marking all possible sequences of tokens up to the maximum phrase length. A sequence is marked if it is known to be the source side of a high confidence phrase translation. This initial segmentation is performed over sequences of tokens where each token is replaced by its word class if it has occurred less than 1000 times in the training data. Using word classes reduces the sparsity in this processing step and is reminiscent of the alignment template approach of Och and Ney (2004).

Unlexicalized preordering model

Since the lexicalized preordering model might run into sparsity issues, we add as a further feature function a weaker model $P_W(\pi \mid h, cs)$. In this model, cs is the set of children represented by their dependency label and whether they are leaves or subtrees and h is the head represented by its part-of-speech tag. This means, the target order is directly predicted based on the source-side dependency labels and the head. The model is estimated via maximum likelihood estimation from the oracle preorderings restricted by the source-side dependency parse tree (*oracle tree reorderings*) over the whole training corpus.

4 Experiments

We perform various experiments to evaluate the ideas presented in the previous section. We will first describe selected details of the implementation of the preordering system and the experimental setup and then provide experimental results and evaluation.

4.1 Implementation and experimental setup

In this section, we give a concise overview of the details of our implementation and the translation setup. Further, we highlight some assumptions and decisions that were necessary for the system training.

Source-side syntax

For source-side reordering to work reliably, the dependency representation must fulfill certain requirements: it should be as *flat* as possible and whenever reasonable, content-bearing elements should be treated as the head. For example, auxiliary verbs should always modify the finite verb and prepositions should be dependents of the head of a noun phrase. We use a customized version of the treebank collection and treebank transformation tool HamleDT (Zeman et al., 2012) for this purpose.

Model training

For training the model, we mostly follow the process from Lerner and Petrov (2013). Training instances are extracted from the automatically aligned training data based on a small set of manually defined rules. To ensure high quality training data, only subtrees that are fully connected by high confidence alignments are considered.

The preordering classifiers are trained on the intersection of high-confidence word alignments and the first-best output of the TurboParser dependency parser (Martins et al., 2009). The alignments are created using the Berkeley aligner[1] with the hard intersection option. The hard intersection option ensures that only high confidence alignment links are produced. While this will lead to a reduction in the number of alignment links, it creates more reliable training data for the preordering model. The dependency parser is trained to produce pseudo-projective dependency trees (Nivre and Nilsson, 2005).[2]

[1] `https://code.google.com/p/berkeleyaligner/`

[2] Projectivization was performed using MaltParser version 1.8; `http://www.maltparser.org/`.

Language models and tuning

During system development, the preordering quality is estimated using Kendall's τ on heldout data. Appropriate values for $k_{+\text{LM}}$ and $k_{-\text{LM}}$ were determined using grid search. We found that beam sizes above $k_{+\text{LM}} = 15$ and $k_{-\text{LM}} = 5$ did not provide any additional improvement in first-best preordering quality.

The set of weights λ for the combination of the preordering model and the language models in the log-linear model are optimized for a selected target metric on heldout data. The straight-forward choice for this metric is Kendall's τ, which indicates the similarity of the the word order of both sides. Kendall's τ is defined as follows (Birch et al., 2010).

$$d_\tau(\pi, \sigma) = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij}}{Z} \tag{13}$$

$$\text{where } z_{ij} = \begin{cases} 1 & \text{if } \pi(i) < \pi(j) \text{ and } \sigma(i) > \sigma(j) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$Z = \frac{(n^2 - n)}{2} \tag{15}$$

The metric indicates the ratio of pairwise order differences between two permutations.

An alternative to the ordering measure is the simulation of a full machine translation system, as first proposed by Tromble and Eisner (2009). To ensure that the changes in word order do not affect this mock translation system and to limit its complexity, the system is limited to phrases of length 1.

Tuning is performed using the *tuning as ranking* (PRO) framework (Hopkins and May, 2011). At tuning time, $k_{-\text{LM}}$ and $k_{+\text{LM}}$ are set to 15 and 100 respectively. *PRO* requires the unweighted values of all feature functions; hence, during tuning time only, we remember the unweighted feature values on each node and sum over intermediate values to arrive at the overall scores. Training instances for ranking are sampled from the best 100 preorderings for each sentence in the tuning set. We perform 6 iterations and interpolate the weights of each iteration with the previous weights by the recommended factor of $\Psi = 0.1$.

We also performed experiments with re-ranking the n-best list of the $-$LM model using the language model. However, as expected, this did not provide better results than the more direct $+$LM integration and we therefore do not report on these results here.

Translation setup

For evaluating the model in a full translation setup, we follow the standard approach to source-side reordering. Given the source side $\mathbf{s}$ and the target side $\mathbf{t}$ of the parallel training corpus, we first perform word alignment using MGIZA++ (Gao and Vogel, 2008). We perform 6 iterations of IBM model 1 training followed by 6 iterations of HMM word alignment and 3 iterations each of IBM model 3 and 4.

After initial training, the preordering model is applied to $\mathbf{s}$, obtaining the preordered corpus $\hat{\mathbf{s}}'$. Since the word order differences between $\hat{\mathbf{s}}'$ and $\mathbf{t}$ should be less acute, less computationally expensive word alignment tools are sufficient to re-align the corpus. We align $\hat{\mathbf{s}}'$ and $\mathbf{t}$ using `fast_align`,[3] an efficient reparameterization of IBM model 2 (Dyer et al., 2013). Improvements in word order can lead to improvements in alignments and hence the training and word alignment process can be performed repeatedly. (Lerner and Petrov, 2013) report no significant improvements after the initial re-alignment. Accordingly, we do not iterate the training process either. The underlying translation system is Moses (Koehn et al., 2007) using the standard feature setup and using only the distortion-based reordering model. Tuning is performed using MERT (Och, 2003). The system is trained on the full parallel sections of the Europarl corpus (Koehn, 2005) and tuned and tested on WMT 2009 and WMT 2010 newstest respectively. The language model is a 5-gram ngram model trained on the target side of Europarl and the news commentary corpus.[4]

[3] `https://github.com/clab/fast_align`

[4] Cf. `http://www.statmt.org/wmt13/translation-task.html`

4.2 Effectiveness of non-local features

While our preliminary results showed that the integration of a language model might be helpful, we now consider this question in more detail. To test whether a language model is beneficial to the reordering model, we compare two versions of the same system: *first-best* −LM is the reordering system without a language model and *first-best* +LM is the same system with the language model integrated via cube pruning. While Kendall's τ gives an impression of the overall word order quality, the BLEU metric gives an indication of the the quality of reorderings within the more restricted space of the length of the ngrams used in the metric. The second row of results show how the quality increases when moving from first-best to *n*-best results.

Model	**Kendall's τ**	**BLEU ($\hat{\mathbf{s}}' \rightarrow \mathbf{s}'$)**
First-best −LM	92.16	68.1
First-best +LM (cube)	92.27	68.7
+LM (cube, oracle 5)	93.33	–
+LM (cube, oracle 10)	93.72	–

Table 1: LM integration tested on first-best prediction (*en–de*).

The results show that the integration of the language model does help the system in the quality of the reorderings. We expected the language model to provide benefits mostly on the borders between tree nodes. The BLEU score indicates an improvement in the ordering of short word sequences, which indicates the presence of this benefit. Next, we examine the quality of the space of preordering provided by the model.

4.3 Quality of the preorderings

Our goal in this work has been to use a syntactic preordering model to delimit the search space for a more complex subsequent model. Hence, in order to examine the model presented in Section 3, we aim to determine the quality of the *n*-best predictions the model makes.

We perform the following experiment for the language pair English–German: Using the preordering system, we produce the 10 best preorderings for each sentence in the test set. We then translate each of the preorderings using a standard phrase-based machine translation system trained on the corpus produced by the first-best preordering system. After translation using a phrase-based system, one translation is selected by an oracle. Table 2 shows results for the oracle criteria (select best translation by sentence-level BLEU score) and for no preordering (baseline). Both systems use a distortion limit of 7 and only the standard distance-based reordering model.

	Distortion	**BLEU**	**MTR**	**TER**
Baseline	7	15.2	35.4	66.6
Oracle ($k = 10$)		17.26	37.97	62.64

Table 2: Estimation of the quality of the *k* best preorderings.

4.4 Discussion

We are interested in several aspects of the output space provided by this system. The first question of interest is whether there are enough good candidates in the space delimited by the preordering system. This question has been answered by the experiments performed in the previous section, which indicate that even within the first 10 best preorderings, enough good instances are contained to enable a significant improvement in translation quality. Since our translation experiments were performed using only automatic evaluation metrics, it is difficult to point out the exact source of the potential improvements we have observed. To examine the gains in more detail and to determine how much the fluency of the output increased, we intend to perform manual evaluation in future work.

The second question of interest is whether the size of the preordering space is manageable for the subsequent model. Since the previous experiments showed that even with only 10 preorderings, a significant improvement can be observed, it is clear that this very small space can be used by a subsequent model. In addition to this, the output in the form of a lattice will allow for using more options and efficient processing using dynamic programming algorithms.

5 Conclusion

Source-side reordering provides a significant potential for translation quality and performance improvement in machine translation, which was shown in previous studies and is further supported by the method's recent surge in popularity. It is therefore an attractive model to extend beyond pure reordering patterns. Most of the benefits of source-side reordering are due to having the ability to model much larger reordering spaces in a more reliable manner than it would be possible within the underlying machine translation system. We propose that this benefit may equally be exploited for other morphosyntactic phenomena such as long-distance agreement. Such phenomena are especially prevalent when translating into morphologically rich languages. These languages continue to pose a multitude of challenges. We aim to address some of these issues with a morphosyntactic adaptation model. As a first step, this paper has explored how a preordering model can be utilized to produce a space of sensible word order predictions, which can in turn be passed to a subsequent model. We have presented a novel preordering model for this purpose and have evaluated its outputs with oracle translation experiments using a common system setup. The results presented here show that a preordering system optimized for producing n-best predictions can provide a valuable output space for further processing.

Acknowledgments

Joachim Daiber is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n^{o} 317471.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations*.

Yehoshua Bar-Hillel, M. Perles, and E. Shamir. 1961. On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14:143–172. Reprinted in Y. Bar-Hillel. (1964). *Language and Information: Selected Essays on their Theory and Application*, Addison-Wesley 1964, 116–150.

Alexandra Birch, Miles Osborne, and Phil Blunsom. 2010. Metrics for mt evaluation: evaluating reordering. *Machine Translation*, 24(1):15–26.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. 2006. Statistical machine reordering. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July. Association for Computational Linguistics.

Adrià de Gispert, Gonzalo Iglesias, and William Byrne. 2015. Fast and accurate preordering for smt using neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2015)*, June.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2011. Quasi-synchronous phrase dependency grammars for machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 474–485, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Teresa Herrmann, Jochen Weiner, Jan Niehues, and Alex Waibel. 2013. Analyzing the potential of source sentence reordering in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2013)*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Laura Jehl, Adrià de Gispert, Mark Hopkins, and Bill Byrne. 2014. Source-side preordering for translation using logistic regression and depth-first branch-and-bound search. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 239–248, Gothenburg, Sweden, April. Association for Computational Linguistics.

Maxim Khalilov and Khalil Sima'an. 2012. Statistical translation after source reordering: Oracles, context-aware models, and empirical analysis. *Natural Language Engineering*, 18:491–519, 10.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Uri Lerner and Slav Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Seattle, Washington, USA, October. Association for Computational Linguistics.

José B Marino, Rafael E Banchs, Josep M Crego, Adria de Gispert, Patrik Lambert, José AR Fonollosa, and Marta R Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Andre Martins, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 342–350, Suntec, Singapore, August. Association for Computational Linguistics.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August. Association for Computational Linguistics.

Yuanbin Wu, Qi Zhang, Xuangjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1541, Singapore, August. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. 2012. Hamledt: To parse or not to parse? In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Latent Domain Word Alignment for Heterogeneous Corpora

Hoang Cuong
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 107, 1098 XG Amsterdam, The Netherlands
c.hoang@uva.nl

Abstract

This paper is fully taken from (Cuong and Sima'an, 2015), which appears at NAACL-HLT 2015.

Word alignment currently constitutes the basis for phrase extraction and reordering in phrase-based systems, and its statistics provide lexical parameters used for smoothing the phrase pair estimates. For over two decades since IBM models (Brown et al., 1993) and the HMM alignment model (Vogel et al., 1996), word alignment remains an active research line, e.g., see recent work (Simion et al., 2013; Tamura et al., 2014; Chang et al., 2014).

During the past years we witnessed an increasing need to collect and use large *heterogeneous* parallel corpora from different domains and sources, e.g., News, Wikipedia, Parliament Proceedings. It is tacitly assumed that assembling a larger corpus should improve a phrase-based system coverage and performance. Recent work (Sennrich et al., 2013; Carpuat et al., 2014; Cuong and Sima'an, 2014b; Kirchhoff and Bilmes, 2014; Cuong and Sima'an, 2014a) shows that this is not necessarily true as phrase translations as well as (bi- and monolingual) word co-occurrence statistics could differ across domains. This suggests that the word alignment quality obtained from IBM and HMM alignment models might also be affected in heterogeneous corpora.

Intuitively, in heterogeneous data certain words are present across many domains, whereas others are more specific to few domains. This suggests that the translation probabilities for words will be as fractioned as the diversity of its translations across the domains. Furthermore, because the IBM and HMM alignment models use *context-insensitive* conditional probabilities, in heterogeneous corpora the estimates of these probabilities will be aggregated over different domains. Both issues could lead to suboptimal word alignment quality.

Surprisingly, the *insensitivity* of the existing IBM and HMM alignment models to domain differences has not received much attention thus far (see the study of Bach et al. (2008) and Gao et al. (2011) for reference in the literature). We conjecture that this is because it is not fully clear how to define what constitutes a *(sub)-domain*. In this paper we propose to exploit the contrast between the alignment statistics in a handful of *seed samples from different domains* in order to induce domain-conditioned probabilities for each sentence pair in the heterogeneous corpus. Crucially, some sentence pairs will be more similar to a seed domain than others, whereas some sentence pairs might be dissimilar to all seed domains. The number and choice of seed domains depends largely on the available resources but intuitively these seed domains are chosen to be relevant to parts of the heterogeneous corpus. A small number of such seeds can be expected to notably improve word alignment accuracy. In fact, a single seed sample already allows us to exploit the contrast between two parts in the corpus: similar or dissimilar to the seed data.

Considering the small seed samples as *partial supervision*, in this paper we explore the question: *how to obtain better word alignment in a heterogeneous, mix-of-domains corpus?* We present a novel

latent domain HMM alignment model, which aims to *tighten* the probability estimates of the generative alignment process of a sentence pair, and of the probability estimates of the sentence pair itself for a specific domain. We also present an accompanying training regime *guided* by partial supervision using the seed samples, exploiting the contrast between the domain-conditioned alignment statistics in these samples. This way we aim for an alignment model that is more domain-sensitive than the original HMM alignment model. Once the domain-conditioned statistics are induced, we discuss how to combine them together to express the probability of a sentence pair as a mixture over specific domains.

Finally, we report experimental results over heterogeneous corpora of 1M, 2M and $4M$ sentence pairs, where we are provided domain information for different samples of 10%, 5% and 2.5% of the heterogeneous data respectively. A large number of experiments are reported, showing that the latent domain HMM model produces notable improvements in word alignment accuracy over the original HMM alignment model. Furthermore, the translation accuracy of the resulting SMT systems is significantly improved across *four* different translation tasks.

1 HMM Alignment Model

In this section, we briefly review the HMM alignment model (Vogel et al., 1996). The generative story of the model is shown in Figure 1. The latent states take values from the target language words and generate source language words.

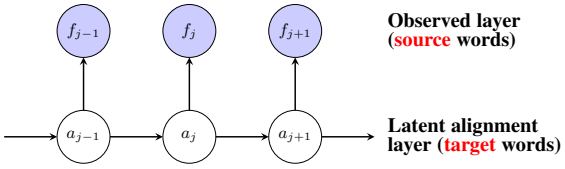


Figure 1: HMM alignment model with observed and latent alignment layers.

Formally, we use $\mathbf{e} = (e_1, \ldots, e_I)$ to denote the target sentence with length I and $\mathbf{f} = (f_1, \ldots, f_J)$ to denote the source sentence with length J. For an alignment $\mathbf{a} = (a_1, \ldots, a_J)$ of a sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$, the model factors $P(\mathbf{f}, \mathbf{a} | \mathbf{e})$ into the word translation and transition probabilities:

$$P(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^{J} P(f_j | e_{a_j}) P(a_j | a_{j-1}). \quad (1)$$

Here, $P(f_j | e_{a_j})$ represents the word translation probabilities and $P(a_j | a_{j-1})$[1] represents the transition probabilities between positions. Note that $P(a_j | a_{j-1})$ depends only on the distance $(a_j - a_{j-1})$. Note also that the first-order dependency model is an extension of the uniform dependency model and zero-order dependency model of IBM models 1 and 2, respectively.

In this work, we model explicitly distances in the range ± 5. Note that *null*-links are also explicitly added in our implementation, following Och and Ney (2003) and Graca et al. (2010).

Once the HMM alignment model is trained, the most probable alignment, $\hat{\mathbf{a}}$ for each sentence pair can be computed by: $\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} P(\mathbf{f}, \mathbf{a} | \mathbf{e})$. Here, the search problem can be solved by the Viterbi algorithm.

2 Latent Domain HMM Alignment Model

Because the heterogeneous data contains a mix of diverse domains, the induced statistics derived from word alignment models reflect translation preferences aggregated over these domains. In this sense, they can be considered domain-*confused* statistics (Cuong and Sima'an, 2014a). This work thus focuses on more **representative** statistics: the domain-*conditioned* word alignment statistics, i.e., the statistics with respect to each of the diverse domains.

[1]The "full" formula for transition probabilities would be $P(a_j | a_{j-1}, I)$. For convenience, we ignore I in our presentation.

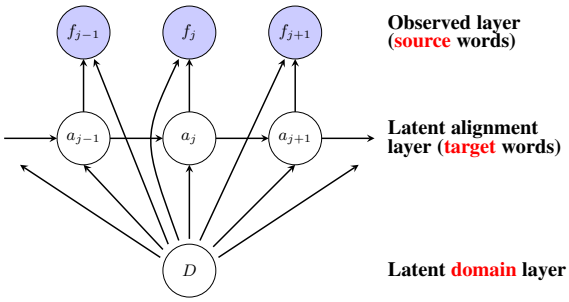


Figure 2: Latent domain HMM alignment model. An additional latent layer representing domains has been conditioned on by both the rest two layers.

By introducing a latent variable D representing domains of the heterogeneous data, we aim to learn the D-conditioned word alignment model $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$.[2] Relying on the HMM alignment model, our latent domain HMM alignment model factors $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ into the domain-conditioned word translation and transition probabilities:

$$P(\mathbf{f}, \mathbf{a}|\mathbf{e}, D) = \prod_{j=1}^{J} P(f_j|e_{a_j}, D)P(a_j|a_{j-1}, D). \tag{2}$$

The generative story of the model is shown in Figure 2. Note how domain-conditioned alignment statistics, $P(\cdot|\ \cdot, D)$ contain their former domain-confused alignment statistics, $P(\cdot|\ \cdot)$ as special case

$$P(f_j|e_{a_j}, D) \propto P(f_j|e_{a_j})P(D|f_j, e_{a_j}), \tag{3}$$
$$P(a_j|a_{j-1}, D) \propto P(a_j|a_{j-1})P(D|a_j, a_{j-1}). \tag{4}$$

With an additional latent domain layer, it becomes crucial to train the model in an efficient way. As suggested by Eq. 3 and 4, we could simplify training by breaking up the estimation process into two steps. That is, we train alignment parameters, $P(\cdot|\ \cdot)$ or domain parameters, $P(D|\ \cdot,\ \cdot)$ first, hold them fixed before training the other kind of the parameters.[3] Instead, in this work we design an algorithm that trains both of them simultaneously via training domain-conditioned parameters $P(\cdot|\ ,\cdot,\ D)$ directly.

2.1 Training

Basically, our model can be viewed as having a set, Θ of N subsets of domain-conditioned parameters, Θ_D for N different domains, i.e., $\Theta = \{\Theta_{D_1},\ \ldots,\ \Theta_{D_N}\}$. In this work, to simplify the learning problem we assume that the domains are very *different* from each other. If this assumption does not hold, the learning problem would shift from *single-label* learning to *multiple-label* learning. We leave this extension for future work.

Our training procedure seeks the parameters Θ that maximize the log-likelihood, $\mathcal{L}$ of the data: $\mathcal{L} = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \log \sum_D \sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{e},\ D,\ \mathbf{a})$. There, however, does not exist a closed-form solution for maximizing $\mathcal{L}$, and EM comes as an alternative solution to fit the model. EM maximizes $\mathcal{L}$ via block-coordinate ascent on a "free energy" lower bound $\mathcal{F}(q,\ \Theta)$ (Neal and Hinton, 1999), using an auxiliary distribution q over both the latent variables: $\mathcal{F}(q,\ \Theta) = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \sum_D \sum_{\mathbf{a}} q \log \frac{P_{\Theta_D}(\mathbf{a},\ D,\ \mathbf{f},\ \mathbf{e})}{q}$.

In the **E**-step of the EM algorithm, we fix Θ and aim to find the distribution q^* that maximizes $\mathcal{F}(q, \Theta)$ over the heterogeneous data. Simple mathematics lead to $\mathcal{F}(q,\ \Theta) = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \log P_{\Theta}(\mathbf{f},\ \mathbf{e}) - KL[q\ ||\ P_{\Theta_D}(\mathbf{a},\ D|\ \mathbf{f},\ \mathbf{e})]$, where $KL[\cdot\ ||\ \cdot]$ is the Kullback-Leiber divergence between two distributions. The distribution q^* can be thus derived as

$$q^* = \frac{P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)}{\sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)} P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e}).$$

[2]Note that $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ contains their former $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})$ as special case, i.e., $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D) = \frac{P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(D|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}{\sum_{\mathbf{f}} \sum_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(D|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}$.

[3]This training scheme is in fact applied in the work of Cuong and Sima'an (2014a), however, for a different purpose.

E-step $\forall D \in \{D_1, \ldots, D_N\}$ do

$$c(D;\ \mathbf{f},\ \mathbf{e}) = P^{(c)}(D|\ \mathbf{f},\ \mathbf{e})$$

$$c(f|\ e;\ \mathbf{f},\ \mathbf{e},\ D) = P^{(c)}(D|\ \mathbf{f},\ \mathbf{e}) \sum\nolimits_{\mathbf{a}} P^{(c)}(\mathbf{a}|\ \mathbf{f},\ \mathbf{e},\ D) \sum\nolimits_{j=1}^{J} \delta(f, f_j) \sum\nolimits_{i=0}^{I} \delta(e,\ e_i)$$

$$c(i|\ i';\ \mathbf{f},\ \mathbf{e},\ D) = P^{(c)}(D|\ \mathbf{f},\ \mathbf{e}) \sum\nolimits_{\mathbf{a}} P^{(c)}(\mathbf{a}|\ \mathbf{f},\ \mathbf{e},\ D) \sum\nolimits_{j=1}^{J} \delta(a_j,\ i)\delta(a_{j-1},\ i')$$

M-step $\forall D \in \{D_1, \ldots, D_N\}$ do

$$P^{(+)}(f|e, D) = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(f|e; \mathbf{f}, \mathbf{e}, D)}{\sum_f \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(f|e; \mathbf{f}, \mathbf{e}, D)}\ P^{(+)}(i|i', D) = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, D)}{\sum_i \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, D)}\ P^{(+)}(D)\ = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(D; \mathbf{f}, \mathbf{e})}{\sum_D \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(D; \mathbf{f}, \mathbf{e})}$$

Figure 3: Pseudocode for the training algorithm for the latent domain HMM alignment model. Note that notation $P^{(c)}$ denotes current iteration estimates, and $P^{(+)}$ denotes the re-estimates.

Here, $P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$ aims to exploit the contrast between the domain-sensitive alignment statistics. Assigning higher probability to one domain forces lower probability assignment to other domains.

Note that $P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ is given in Eq. 2 and $\sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ can be computed efficiently using dynamic programming.[4] Meanwhile, $P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$ can be derived by Bayes' rule, i.e.,

$$P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e}) \propto P_{\Theta_D}(\mathbf{f},\ \mathbf{e}|\ D) P_{\Theta_D}(D).$$

Here, the estimation of the domain prior parameters is easy, $P_{\Theta_D}(D) \propto \sum_{\langle \mathbf{f},\ \mathbf{e} \rangle} P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$. The estimation of $P_{\Theta_D}(\mathbf{f},\ \mathbf{e}|\ D)$ raises a task of defining a generative process for every sentence pair in the heterogeneous data with respect to a specific domain. Following (Cuong and Sima'an, 2014b), we factor it into two kinds of models in a symmetrized strategy: $P_{\Theta_D}(\mathbf{f},\ \mathbf{e}|\ D) \propto \big(P_{\Theta_D}(\mathbf{e}|\ D) P_{\Theta_D}(\mathbf{f}|\ \mathbf{e},\ D) + P_{\Theta_D}(\mathbf{f}|\ D) P_{\Theta_D}(\mathbf{e}|\ \mathbf{f},\ D)\big)$.

Basically, $P_{\Theta_D}(\cdot|\ \cdot,\ D)$ can be thought of as the domain-conditioned translation models, aiming to model how well a target/source sentence is generated over a source/target sentence with respect to a domain.[5] Meanwhile, $P_{\Theta_D}(\cdot|\ D)$ can be thought of as the domain-conditioned language models (LMs), aiming to model how fluent a source/target sentence with respect to a domain. For simplicity, once the domain-conditioned LMs are trained, they will stay *fixed* during training, i.e., LM probabilities are not parameters in our model.

In the **M**-step of the EM algorithm, we fix the derived q^* and aim to find the parameter set Θ^* that maximizes $\mathcal{F}(q, \Theta)$ over the data. This can be (easily) done by using q^* to softly fill in the values of $\mathbf{a}$ and D to estimate model parameters.

Pseudocode

In summary, the model has three kinds of parameters - word translation, word transition, and domain prior parameters. We now summarize the training via presenting the pseudocode.

First, we present expected count notations with respect to domains for the parameters. We use $c(f|\ e;\ \mathbf{f},\ \mathbf{e},\ D)$ to denote the expected counts that word e aligns to word f. We use $c(i|\ i';\ \mathbf{f},\ \mathbf{e},\ D)$ to denote the expected counts that two certain consecutive source words j and $j-1$ align to two target words i and i' respectively, i.e., j aligns to i and $j-1$ aligns to i'. Finally, we also use $c(D;\ \mathbf{f},\ \mathbf{e})$ to denote the expected count of domain priors. Note that all the expected counts are in the translation $(\mathbf{f}|\ \mathbf{e})$.

Figure 3 represents the pseudocode.

3 Learning with Partial Supervision

We now discuss remaining issues on how to guide the learning with partial supervision, i.e., how to use the given domain information of seed samples to guide the learning.

Number of Domains The values of $D \in [1..(N+1)]$ depends on the N available seed samples plus the

[4] Its time complexity is $\mathcal{O}(J \times I^2)$ for each sentence pair $\langle \mathbf{f},\ \mathbf{e} \rangle$ with their length J and I respectively.

[5] Note that $P_{\Theta_D}(\cdot|\ \cdot,\ D) = \sum_{\mathbf{a}} P_{\Theta_D}(\cdot,\ \mathbf{a}|\ \cdot,\ D)$ and it can be thus computed efficiently using dynamic programming.

so-called "out-domain," i.e., the part of the heterogeneous data that is dissimilar to all of the N sample domains.

Parameter Initialization We first discuss how to initialize the domain prior parameters. If a sentence pair $\langle \mathbf{f},\ \mathbf{e} \rangle$ belongs to a sample with a pre-specified domain D_i, we initialize $P(D_i|\ \mathbf{f},\ \mathbf{e})$ close to 1, and, $P(D_{i'}|\ \mathbf{f},\ \mathbf{e})$ close to 0 for other domains i', $i' \neq i$. Furthermore, we uniformly create the domain prior parameters for the rest of sentence pairs.

Uniform initialization for the domain-conditioned alignment parameters is also a reasonable option. Nevertheless, a more effective way is to make use of the domain-specific seed samples and the pool of the rest sentence pairs in the heterogeneous data.[6] That is, we train the model on each of the samples, assigning the derived probabilities as the initialization for their corresponding domain-conditioned alignment parameters. In our implementation, one EM iteration is usually dedicated for this. It should be noted that we ignore the domain prior parameters in the model during the period.

Parameter Constraints During training, it would be also necessary to keep the domain prior parameters fixed for all sentence pairs that belong to seed samples. This can be thought of as the constraints derived from the partial knowledge, guiding the learning to a desirable parameter space.

Domain-conditioned LMs training We now discuss how to train the domain-conditioned LMs with partial supervision. It would be reasonable to use the domain-specific seed samples to train their exemplifying domain-conditioned LMs, and the pool of the rest sentence pairs to train the out-domain LMs. Nevertheless, the out-domain LMs trained on such a big corpus could dominate the other domain-conditioned LMs. Following Cuong and Sima'an (2014b), we rather create a "pseudo" out-domain sample to train the out-domain LMs, i.e., the creation is via an inspired burn-in period. In brief, an EM iteration is dedicated just to compute $P(D_{OUT}|\ \mathbf{f},\ \mathbf{e})$ for all sentences, ranking them and select a small subset with highest score as the (on the fly) pseudo out-domain sample.

Note that our partial learning framework is very simple. There are various advanced learning frameworks that are also applicable with the partial supervision, e.g., Posterior Regularization (Ganchev et al., 2010). This leaves much space for future work.

4 Domain-conditioned Decoding

At test time, assigning each sentence pair to a single most likely domain (hard decision) is likely to result in sub-optimal performance.[7] Instead we average over domains (soft decision) while predicting the translation. Formally for each sentence pair, $\langle \mathbf{e},\ \mathbf{f} \rangle$, we can find their best Viterbi alignment, $\hat{\mathbf{a}}$ as follows:

$$\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} \sum\nolimits_D P(\mathbf{f}, \mathbf{a}|\mathbf{e}, D)P(\mathbf{e}|D)P(D).$$

Here, we derive the last equation by applying Bayes' rule to $P(D|\ \mathbf{e})$, i.e., $P(D|\ \mathbf{e}) \propto P(\mathbf{e}|\ D)P(D)$. Interestingly, our Viterbi decoding now relies on a mix of domain-conditioned statistics for each sentence pair. The computing of term $\sum_D(\mathbf{a})$ for all possible alignments, $\mathbf{a}$, however, is intractable, making the search problem difficult. Inspired by Liang et al. (2006), we opt instead for a heuristic objective function as follows[8]:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmax}} \prod\nolimits_D P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}, D)^{P(\mathbf{e}|\ D)P(D)}. \quad (5)$$

Here, note that $\prod p$ is a lower bound for $\sum p$, when $0 \leq p \leq 1$, according to Jensen's inequality. With Eq. 5, it is straightforward to design a dynamic programming algorithm to decode, e.g., the Viterbi

[6] During the initialization, we assume that the pool of the rest sentence pairs in the heterogeneous data is the exemplifying sample of the out-domain.

[7] Later experiments on word alignment will confirm this.

[8] Alternative solutions could be Lagrangian relaxation-based decoder (DeNero and Macherey, 2011; Chang et al., 2014).

algorithm. In practice, we observe that the approximation yields good results. Later experiments on word alignment will present this in detail.

5 Experimental Setup

Model	Domain Prior	Prec.↑	Δ	Rec.↑	Δ	AER↓	Δ
		1 **Million**					
Model 4 (ref.)	-	71.56	-	64.59	-	32.10	-
Baseline	-	66.95	-	61.29	-	36.00	-
Latent	Pharmacy	67.85	**+0.90**	61.72	**+0.43**	35.36	**-0.64**
	Legal	67.57	**+0.62**	62.29	**+1.00**	35.17	**-0.83**
	Hardware	69.41	**+2.46**	63.58	**+2.29**	33.63	**-2.37**
	Legal + Hardware + Pharmacy	**69.64**	**+2.69**	**63.30**	**+2.01**	**33.68**	**-2.32**
		2 **Million**					
Model 4 (ref.)	-	74.13	-	65.30	-	30.56	-
Baseline	-	68.34	-	61.58	-	35.22	-
Latent	Pharmacy	68.85	**+0.51**	62.58	**+1.00**	34.43	**-0.79**
	Legal	69.98	**+1.64**	64.01	**+2.43**	33.13	**-2.09**
	Hardware	69.45	**+1.11**	63.23	**+1.65**	33.81	**-1.41**
	Legal + Hardware + Pharmacy	**71.51**	**+3.17**	**63.87**	**+2.29**	**32.53**	**-2.69**
		4 **Million**					
Model 4 (ref.)	-	75.53	-	65.95	-	29.58	-
Baseline	-	69.37	-	64.30	-	33.26	-
Latent	Pharmacy	69.69	**+0.32**	62.80	-1.50	33.94	+0.68
	Legal	70.51	**+1.14**	63.94	-0.36	32.93	**-0.33**
	Hardware	71.75	**+2.38**	64.44	**+0.14**	32.10	**-1.16**
	Legal + Hardware + Pharmacy	**72.16**	**+2.79**	**64.30**	±0.0	**31.99**	**-1.27**

Table 1: Alignment accuracy over heterogeneous corpora.

In the following experiments, we use three heterogeneous English-Spanish corpora consisting of $1M$, $2M$ and $4M$ sentence pairs respectively. These corpora combine two parts. The first part respectively $0.7M$, $1.7M$ and $3.7M$ is collected from multiple domains and resources including EuroParl (Koehn, 2005), Common Crawl, United Nation, News Commentary. The second part consists of three domain-exemplifying samples consisting of roughly $100K$ sentence pairs for each one (total $300K$). Each of these three samples (manually collected by a commercial partner) exemplifies a specific domain related to **Legal**, **Hardware** and **Pharmacy**.

Outlook In Section 6 we examine the word alignment yielded by the HMM alignment model and our latent domain HMM alignment model. In Section 7 we proceed further to examine the translation produced by derived SMT systems.

6 Word Alignment Experiment

For alignment accuracy evaluation, we use a data set of 100 sentence pairs with their "golden" alignment from Graca et al. (2008). Here, the golden alignment consists of *sure* links (S) and *possible* links (P) for each sentence pair. Counting the set of generating *alignment* links (A), we report the word alignment accuracy by *precision* ($\frac{|A \cap P|}{|P|}$), *recall* ($\frac{|A \cap S|}{|S|}$), *alignment error rate* (AER) ($1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$) (Och and Ney, 2003).[9]

For all experiments, we use the same training configuration for both the baseline/the latent domain alignment model: 5 iterations for IBM model 1/the latent domain model; 3 iterations for HMM alignment model/the latent domain model. For evaluation, we first align the sentence pairs in both directions and then symmetrize them using the *grow-diag-final* heuristic (Koehn et al., 2003).

For reference we also report the performance of a considerably more expressive Model 4, capable of capturing more structure, but at the expense of intractable inference. Using MGIZA++ (Gao and Vogel,

[9]Note that better results correspond to larger Precision, Recall and to smaller AER.

2008), we run 5 iterations for training Model 1, 3 iterations for training the HMM alignment model, Model 3 and Model 4.

6.1 Learning with Single Domain

We first examine the binary case, where we are given domain information in advance for each kind of samples **only**, e.g., Legal, or Pharmacy, or Hardware. For the different sizes of the heterogeneous data ($1M$, $2M$ and $4M$) the seed sample size is thus 10%, 5% and 2.5% respectively. Note that in such cases, training the latent domain alignment model induces two domain-conditioned statistics: in-domain vs. out-domain (D_1 and D_2 respectively). Once the model is trained, we combine the induced domain-conditioned statistics together (Eq. 5) and examine the produced word alignment output.

Table 1 presents the results. Most importantly, it shows that as long as providing domain information for reasonably large enough data, learning the latent domain alignment model notably improves the word alignment accuracy. For instance, given in advance the domain information for a sample of 10%, and 5% of the heterogeneous corpora, our model consistently improves the word alignment accuracy in all cases. Meanwhile, given in advance the domain information for a relatively small sample of 2.5% of the heterogeneous data, the results are mixed. We obtain a good performance/slightly better performance/worse performance with the case of Hardware/Legal/Pharmacy respectively.

What do domain-conditioned statistics look like?

To have an idea what the induced statistics look like, we investigate their **conditional entropy**. Here, we present the conditional entropy for the domain-confused/-conditioned word translation statistics induced from the HMM alignment model/its latent domain model. Note that similar results are observed for transition tables.

Model	Prior	Statistics	H(F| E)
Baseline	-	Domain-confused	**1348.53**
Latent	Hardware	D_1-conditioned	**1124.43**
		D_2-conditioned	**1354.58**
	Legal	D_1-conditioned	**1104.58**
		D_2-conditioned	**1385.35**
	Pharmacy	D_1-conditioned	**1115.52**
		D_2-conditioned	**1342.54**

Table 2: Conditional entropy of the statistics.

Formally, for a translation table, $\langle F,\ E\rangle$, its conditional entropy, $H(F|\ E)$ can be estimated from its possible word pairs, $\langle e,\ f\rangle$: $H(F|\ E) = -\sum_e P(e) \sum_f P(f|\ e) \log P(f|\ e)$. Table 2 reveals that the induced D_1-conditioned statistics need much less *bits* to represent than the induced domain-confused statistics, e.g., 1124.43, 1104.58, 1115.52 vs. 1348.53. This implies the induced D_1-conditioned statistics are much more **predictable** compared to the domain-confused statistics. Meanwhile, the induced D_2-conditioned statistics are similar to the domain-confused statistics in terms of the conditional entropy, e.g., 1354.58, 1385.35, 1342.54 vs. 1348.53.

6.2 Learning with Multiple Domains

It would be more interesting to learn the latent domain alignment model for multiple domains, rather than learning with each of them separately. In detail, using **all** the seed samples from different domains, we aim to learn four different domain-conditioned statistics simultaneously. Under this setting, we obtain good results, as described in Table 1. For the two cases with the training corpora of 2M and 4M sentence pairs respectively, learning with the combining domain prior knowledge produces the best word alignment accuracy compared to the rest. In the last case with the training corpus of 1M sentence pairs, learning with the combining domain prior knowledge produces compatible with the case of Hardware, i.e., the best binary domain case.

Table 1 also reveals that the performance of our model approaches Model 4, even though Model 4 is much more complex and computationally expensive.

Domain-conditioned statistics combination

We also investigate the relation between the number of domain-conditioned statistics "involved" in the Viterbi decoding (Eq. 5) and the word alignment accuracy. Table 3 presents the results in case of using only the induced D_1-/, D_2-/, D_3-/, D_4-conditioned statistics separately, and also using their different combinations. Interestingly, we observe that using more domain-conditioned statistics for decoding incrementally improves the word alignment accuracy over the heterogeneous data. While the domain-conditioned statistics are very different in their characteristics from each other, the results reveal how they are *complementary* to the others, conveying a mix of domains for each sentence pair.

Decoding's Statistics	**Prec.↑**	**Rec.↑**	**AER↓**
Hard Decision (ref.)	68.49	62.80	34.48
D_1 (Pharmacy)	64.78	59.86	37.78
D_2 (Legal)	66.54	61.15	36.27
D_3 (Hardware)	66.98	61.36	35.95
D_4 (OUT)	68.46	63.01	34.38
D_1 + D_2	66.80	61.72	35.84
D_1 + D_2 + D_3	68.54	62.80	34.46
D_1 + D_2 + D_3 + D_4	**69.64**	**63.30**	**33.68**

Table 3: Domain-conditioned statistics combination for Viterbi decoding. The reported results are for the heterogeneous corpus of 1M sentence pairs. Similar results are observed for other training data.

Finally, it is also tempting to make a comparison between the *hard* vs. *soft* domain assignment in Viterbi decoding. Here, for hard domain decision we simply do decoding with the following objective function: $\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}, \hat{D})$, where $\hat{D} = \text{argmax}_D P(D|\ \mathbf{e})$. Table 3 presents the results. It reveals that a soft domain assignment on the domain of sentence pairs results in a better alignment accuracy than a hard domain assignment.[10]

7 Translation Experiment

In this section, we investigate the contribution of our model in terms of the translation accuracy. Here, we run experiments on the heterogeneous corpora of 1M, 2M, and 4M sentence pairs, testing the translation accuracy over four different domain-specific test sets related to News, Pharmacy, Legal, and Hardware.

We use a standard state-of-the-art phrase-based system as the baseline. Our dense features include MOSES (Koehn et al., 2007) baseline features, plus hierarchical lexicalized reordering model features (Galley and Manning, 2008), and the word-level feature derived from IBM model 1 score, c.f., (Och et al., 2004).[11] The interpolated 5-grams LMs with Kneser-Ney are trained on a very large monolingual corpus of 2B words. We tune the systems using k-best batch MIRA (Cherry and Foster, 2012). Finally, we use MOSES (Koehn et al., 2007) as decoder.

Our system has exactly the same setting with the baseline, except: (1) To learn the translation, we use the alignment result derived from our latent domain HMM alignment model, rather than the HMM alignment model; and (2) We replace the word-level feature with our four domain-conditioned word-level features derived from the latent domain IBM model 1. Here, note that our latent model is learned with the supervision from the combining domain knowledge of all three domain-specific seed samples.

[10]Note that similar results are also observed for training, in which a soft domain assignment using soft EM produces better alignment accuracy than a hard domain assignment using hard EM. (See (Gao et al., 2011) for reference to hard domain assignment to training data.) This is perhaps due to the characteristics of the data we use. For instance, News sentence pairs are useful for translating Legal, Financial or EuroParl to varying degrees.

[11]For every phrase pair $\langle \tilde{f},\ \tilde{e} \rangle$ with their length of $m_{\tilde{f}}$ and $l_{\tilde{e}}$ respectively, the lexical feature estimates a probability in Model 1 style between their word pairs $\langle f_j,\ e_i \rangle$ (i.e. $P(\tilde{f}|\ \tilde{e}) = \frac{\epsilon}{l_{\tilde{e}}} \prod_{j=1}^{m_{\tilde{f}}} \sum_{i=1}^{l_{\tilde{e}}} P(f_j|e_i)$). Note that adding word-level features from both translation sides does not help much, as observed by (Och et al., 2004). We thus add only an one from a translation side.

Data	System	BLEU↑	METEOR↑	TER↓
News test				
1M	Model 4 (ref.)	23.6	30.8	58.3
	Baseline	23.2	30.6	58.9
	Our System	23.5/**+0.3**	30.8/+0.2	58.7/-0.2
2M	Baseline	25.9	32.4	56.1
	Our System	26.3/**+0.4**	32.6/**+0.2**	55.6/-**0.5**
4M	Baseline	26.8	33.0	55.0
	Our System	27.0/+**0.2**	33.1/+0.1	54.7/-**0.3**
Pharmacy				
1M	Model 4 (ref.)	54.7	43.8	33.4
	Baseline	53.9	43.4	34.6
	Our System	54.4/+**0.5**	43.8/**+0.4**	34.0/**-0.6**
2M	Baseline	54.5	43.7	34.4
	Our System	55.3/**+0.8**	44.3/**+0.6**	33.5/**-0.9**
4M	Baseline	54.8	43.9	33.8
	Our System	55.0/**+0.2**	44.0/+0.1	33.7/-0.1
Legal				
1M	Model 4 (ref.)	56.6	44.7	34.1
	Baseline	56.0	44.2	35.0
	Our System	57.2/**+1.2**	44.4/+0.2	34.0/**-1.0**
2M	Baseline	55.8	43.9	35.4
	Our System	58.3/**+2.5**	44.7/**+0.8**	**33.4**/**-2.0**
4M	Baseline	55.9	43.9	34.3
	Our System	57.3/**+1.4**	44.4/**+0.5**	33.4/**-0.9**
Hardware				
1M	Model 4 (ref.)	75.4	53.6	17.7
	Baseline	74.9	53.1	19.0
	Our System	76.8/**+1.9**	53.9/**+0.8**	17.3/**-1.7**
2M	Baseline	75.7	53.5	18.6
	Our System	77.4/**+1.7**	54.3/**+0.8**	17.0/**-1.6**
4M	Baseline	77.1	54.2	17.3
	Our System	77.9/**+0.8**	54.5/**+0.3**	16.7/**-0.6**

Table 4: Metric scores for the systems, which are averages over multiple runs. Bold results indicate that the comparison is significant over the baseline.

For the News translation task, we tune systems on the News-test 2008 of $2,051$ sentence pairs and test them on the News-test 2013 of $3,000$ sentence pairs from the WMT 2013 shared task (Bojar et al., 2013). For the Pharmacy, Legal, and Hardware translation tasks, we tune systems on three domain-specific dev sets of $1,000$ sentence pairs and test them on three domain-specific test sets of $1,016$, $1,326$ and $1,721$ sentence pairs. We report three metrics - BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011) and TER (Snover et al., 2006), with statistical significance at 95% confidence interval under paired bootstrap re-sampling.[12] For every system reported, we run the optimizer three times, before running MultEval (Clark et al., 2011) for resampling and significance testing.

Data	BLEU↑	METEOR↑	TER↓
1M	**+1.0**	**+0.4**	**-0.9**
2M	**+1.4**	**+0.6**	**-1.3**
4M	**+0.7**	**+0.3**	**-0.5**

Table 5: Averaged improvements across the tasks.

Results are in Table 4, showing significant improvements across four different test sets over different heterogeneous corpora sizes. Table 5 gives a summary of the improvements. On average, over heterogeneous corpora of 1M, 2M and 4M sentence pairs, our system outperforms the baseline by 1.0 BLEU, 1.4 BLEU and 0.7 BLEU, respectively. Finally, we observe that our system produces comparably

[12]Note that better results correspond to larger BLEU, METEOR and to smaller TER.

good performance to the MGIZA++-based system. When 1M data is considered, on *three* of *four* tasks, our system produces at least compatible translation accuracy to the corresponding MGIZA++-based system.

Further analysis reveals that the improvement is due to not only the reduction in alignment error rate, but also the use of the domain-sensitive lexical features. Moreover, the domain-sensitive lexical features is particularly useful when the domain of the test data matches with the domain of seed samplers. This is also widely observed in the literature, e.g., see (Eidelman et al., 2012; Hasler et al., 2014; Hu et al., 2014).

8 Related Work and Conclusion

In terms of domain-conditioned statistics for word alignment, a distantly related research line (Tam et al., 2007; Zhao and Xing, 2008) focuses on using document topics to improve the word alignment. In terms of learning word alignment with partial supervision, another distantly related research line focuses on semi-supervised training with partial manual alignments (Fraser and Marcu, 2006; Gao and Vogel, 2010; Gao et al., 2010). Finally, recent work also focuses on data selection (Kirchhoff and Bilmes, 2014; Cuong and Sima'an, 2014b), mixture models (Carpuat et al., 2014), instance weighting (Foster et al., 2010) and latent variable models (Cuong and Sima'an, 2014a) over heterogeneous corpora.

One main contribution of this work is the novelty of exploring the quality of word alignment in heterogeneous corpora. This, surprisingly, has not received much attention thus far (see the study of Bach et al. (2008) and Gao et al. (2011) for reference in the literature). Another major contribution of this work is a learning framework for latent domain word alignment with partial supervision using seed domains. We present its benefits for improving not only the word alignment accuracy, but also the translation accuracy resulting SMT systems produce. We hope this study sparks a new research direction for using domain samples, which is cheap to gather, but has not been exploited before.

One obvious direction for future work might be to integrate the model into fertility-based alignment models (Brown et al., 1993), as well as other recently advanced alignment frameworks, e.g., (Simion et al., 2013; Tamura et al., 2014; Chang et al., 2014). Another interesting direction might be to integrate our model into advanced mixing multiple translation models, improving SMT systems trained on the heterogeneous data (Razmara et al., 2012; Sennrich et al., 2013; Carpuat et al., 2014). Finally, an open question is whether it is possible to learn the latent domain alignment model in a fully unsupervised style. This challenge deserves more attention in future work.

Acknowledgements

The first author is supported by the EXPERT (EXPloiting Empirical appRoaches to Translation) Initial Training Network (ITN) of the European Union's Seventh Framework Programme.

References

Nguyen Bach, Qin Gao, and Stephan Vogel. 2008. Improving word alignment with language model based confidence scores. In *Proceedings of the Third Workshop on Statistical Machine Translation*.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June.

Marine Carpuat, Cyril Goutte, and George Foster. 2014. Linear mixture models for robust machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Yin-Wen Chang, Alexander M. Rush, John DeNero, and Michael Collins. 2014. A constrained viterbi relaxation for bidirectional word alignment. In *Proceedings of ACL*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of NAACL HLT*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of HLT: Short Papers*.

Hoang Cuong and Khalil Sima'an. 2014a. Latent domain phrase-based models for adaptation. In *Proceedings of EMNLP*.

Hoang Cuong and Khalil Sima'an. 2014b. Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING*.

Hoang Cuong and Khalil Sima'an. 2015. Latent domain word alignment for heterogeneous corpora. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, USA, May.

John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In *Proceedings of ACL*.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of ACL: Short Papers*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of EMNLP*.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of ACL*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP*.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049, August.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08.

Qin Gao and Stephan Vogel. 2010. Consensus versus expertise: A case study of word alignment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10.

Qin Gao, Nguyen Bach, and Stephan Vogel. 2010. A semi-supervised word alignment algorithm with partial manual alignments. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10.

Qin Gao, Will Lewis, Chris Quirk, and Mei-Yuh Hwang. 2011. Incremental training and intentional over-fitting of word alignment. In *Proceedings of MT Summit XIII*.

Joao Graca, Joana Paulo Pardal, Luisa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Joao Graca, Kuzman Ganchev, and Ben Taskar. 2010. Learning tractable word alignment models with complex constraints. *Comput. Linguist.*, 36(3):481–504.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based mt. In *Proceedings of EACL*.

Yuening Hu, Ke Zhai, Vladimir Eidelman, and Jordan Boyd-Graber. 2014. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of ACL*.

Katrin Kirchhoff and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *Proceedings of EMNLP*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, MMichigan0605 AAMT.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*.

Radford M. Neal and Geoffrey E. Hinton. 1999. Learning in graphical models. chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pages 355–368. MIT Press.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of ACL*.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of ACL*.

Andrei Simion, Michael Collins, and Cliff Stein. 2013. A convex alternative to ibm model 2. Proceedings of EMNLP.

Matthew Snover, Bonnie Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual lsa-based adaptation for statistical machine translation. *Machine Translation*, 21(4):187–207.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of ACL*.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*.

Bing Zhao and Eric P. Xing. 2008. Hm-bitam: Bilingual topic exploration, word alignment, and translation. In *Proceedings of NIPS*.

Semantic Textual Similarity in Machine Translation Evaluation

Hanna Béchara
Research Group in
Computational Linguistics
University of Wolverhampton
hanna.bechara
@wlv.ac.uk

Abstract

Comparative assessment of machine translation tools often relies on automatic metrics to evaluate systems. However, most automatic metrics still fail to correlate to human judgement. In an attempt to focus further on the adequacy and informativeness of translations, we integrate features of semantic similarity into the evaluation process. By using methods previously employed in semantic textual similarity (STS) tasks, we use semantically similar sentences and their quality scores in order to estimate the quality of machine translated sentences. Our results show that this method can improve the prediction of machine translation quality for semantically similar sentences.

1 Introduction

Since the introduction of statistical machine translation (SMT) in 1990 (Brown et al., 1990), data-driven research, in particular phrase based statistical machine translation (PB-SMT), has become the most dominant strand of research in machine translation (MT). Statistical machine translation systems rely on statistical models trained on existing parallel corpora and work best when significant amounts of training data (i.e. parallel bilingual corpora) are available for the language pair. While statistical systems can produce somewhat erratic results, they tend to be more robust than their rule-based predecessors (**?**).

In the context of these advances in machine translation tools, comparative assessment of the various outputs is a challenging yet important part of the process. Developers have turned to a variety of techniques to assess the quality of machine translation output. Today, human evaluation is still considered the best and most reliable judgement in machine translation evaluation. However, this method is inefficient, especially when large corpora are involved. Automatic evaluation metrics have been developed to evaluate MT output quality, but these rely on reference translations and focus mainly on syntactic and surface similarities, rather than semantic accuracy. Furthermore, quality estimation and machine learning techniques have become the focus of MT output evaluation, as they can be used to measure different aspects of correctness. One aspect of correctness that has not been subject of enough research is the notion of semantic correctness. While several tools that measure monolingual similarity have been developed, the extent to which such tools can help in machine translation evaluation across languages has not been fully researched.

This paper addresses the use of semantic correctness in evaluation by integrating semantic textual similarity measures into the evaluation process, without relying on a reference translation. The rest of this paper is structured as follows: Section 2 takes a look at a few previous attempts to integrate semantic similarity into evaluation. Section 3 details the data and tools used during the course of this research. Section 4 describes our approach and details a series of experiments on different datasets that investigates the effective use of semantic information as a tool for evaluation. Section 5 presents the conclusions we draw from the results of our experiments.

2 Related Work

Developers rely on a variety of techniques to assess the quality of machine translation output. While human evaluation is still the best and most reliable assessment measure, it is also costly and time-

consuming as it requires hours of long manual assessment, often by highly trained translators. This renders it inefficient in the context of larger corpora. Automatic evaluation metrics have been developed to estimate the quality of MT output. Automatic evaluation metrics compare MT output to a reference translation, which is a translation provided by a human and considered to be a "gold standard" translation. The assumption is that the score returned would mimic human judgement, as the closer the output is to the human "gold standard", the higher its quality. BLEU (Papineni et al., 2002) is a popular and widely used automatic metric that relies on n-gram overlapping to approximate human judgements. BLEU matches *n*-grams between the MT output and the reference translation, using *n*-gram precision with a brevity penalty as the score. Criticisms of BLEU and n-gram matching metrics in general are addressed by Callison-Burch et. al. (2008), who show that BLEU fails to correlate to (and even contradicts) human judgement. BLEU is very sensitive to small changes in the output, and fails to capture linguistic variations, especially in the case where only one reference translation is being used. Furthermore, metrics such as BLEU are specifically designed for corpus-level assessment, and do not fare well when evaluating quality on a sentence-level. Additionally, these metrics face serious limitations in that they rely heavily on reference translations, limiting their flexibility. If an automatic translation fails to match a given reference translation, it will be penalised by the metric even if it is a fully fluent and adequate translation. While multiple references mitigate this problem somewhat, it is highly impractical to cover every single possible translation for a given input. Furthermore, not all resources will have access to a reference translation.

To date, relatively few attempts have been made to use semantic information for MT evaluation. Gimènez and Màrquez (2007) propose metrics which take linguistic features at more abstract levels into account. They show that metrics based on deeper linguistic information make up for the short-comings of automatic evaluation metrics and produce more reliable system rankings that better correlate with human judgement. Their metric is based on shallow semantic structures such as word forms, part of speech tags, dependency relationships, syntactic phrases semantic roles and named entities. They call these structures linguistic elements (LE), and posit that a sentence can be seen as a bag of linguistic elements. Their system outperforms metrics based on lexical matching alone. However, they find that semantic oriented metrics are more stable at system level rather than at sentence level.

More recently, Lo and Wu (2011) argue that reference-based metrics such as BLEU (Papineni et al., 2002) do not adequately capture semantic correctness between the machine translation output and the reference translation. They define a good translation as as one that preserves the central information, rather than focusing on fluency. They present their alternative, MEANT, a semi-automatic metric that assesses translations by matching semantic role fillers. MEANT, however, is semi-automatic, as it relies on human judgement to determine the correctness of these semantic role-fillers. However, it is more efficient and less labour-intensive than pure manual evaluation.

Castillo and Estrella (2012) follow in this line of research, claiming that the output of machine translation systems will correlate more strongly with human translations if they have a higher semantic textual similarity score with the reference translation. Using a machine learning approach based on 8 sentence-level semantic features, they determine a semantic similarity score between each output segment and its corresponding reference translation. They report competitive scores at system-level.

While these metrics show some level of success, they still rely on the existence of a reference translation to which to compare the output.

The restrictions and short-comings of the reference-based translation metrics have led into further investigation of the evaluation problem. Reference-free evaluation has stepped in to address the problems introduced by the need for a reference translation. Early work in quality estimation built on the concept of confidence estimation used in speech recognition. These systems usually relied on system-dependent features, and focused on measuring how confident a given system is rather than how correct the translation is. Later experiments in quality estimation used only system-independent features based on the source sentence and target translation (Specia et al., 2009b). They train an SVM regression model based on 74 shallow features, and report significant gains in accuracy over MT evaluation metrics. At first, these approaches to quality estimation focused mainly on shallow features. Such features include

n-gram counts, the average length of tokens, punctuation statistics, sentence length. Later systems incorporate linguistic features such as part of speech tags, syntactic information and word alignment information.

The main advantage of using quality estimation is that it requires no reference translation to predict quality. Furthermore, the term ”quality” itself is flexible, and can change to reflect specific applications, from quality assurance, estimating post-editing effort and ranking translations. Specia et al. (2009a) define quality in terms of post-editing efficiency, using quality estimation to filter out sentences that would require too much time to post-edit. Similarity, He et al (2010) use quality estimation to predict human post-editing effort and recommend the SMT outputs to a translation memory user based on estimated post-editing effort. However, Specia, Raj and Turchi (2010) use quality estimation to rank translations from different systems and highlight inadequate segments.

We propose a method to evaluate semantic similarity in machine translation output where a reference translation is not available, using machine learning techniques similar to those used in quality estimation. As semantic textual similarity tools are widely monolingual, it is not possibly to compare the output directly to the source. Like Castillo and Estrella (2012), we use semantic textual similarity techniques in order to determine whether or not the machine translation output preserves the meaning of the original sentence. However, in contrast to their method, we do not rely on a reference translation.

3 Data and Tools

The experiments we propose require sentences with a certain degree of semantic similarity, and a way to measure this similarity. We also require machine translated output to evaluate, preferably with a certain measure of quality (a gold standard or a reference translation) to which we can compare our system. We use a number of open source tools and freely available corpora to design and test our experiments. However, as these tools do not always fulfil our purpose, we use a number of tools and datasets of our own design as well. This section outlines both the data and tools used in our research.

3.1 Parallel Corpora

We used a variety of pre-existing freely-available corpora to train and test our systems, in addition to a dataset of our own design. This section sheds some light on these datasets.

3.1.1 The SICK Dataset

SICK (Sentences Involving Compositional Knowledge) is a dataset specifically for compositional distributional semantics. It includes a large number of English sentence pairs that are rich in lexical, syntactic and semantic phenomena. The SICK dataset is generated from existing datasets based on images and video descriptions, and each sentence pair annotated for relatedness (similarity) and entailment by means of crowd-sourcing techniques (Marelli et al., 2014b). The similarity score is a continuous score between 1 and 5, further defined in Table 1. For our purpose, we extract 5,000 sentence pairs to use in our experiments.

Table 1: Semantic Textual Similarity scale used by SemEval

0	The two sentences are on different topics
1	The two sentences are not equivalent, but are on the same topic
2	The two sentences are not equivalent, but share some details
3	The two sentences are roughly equivalent, but some important information differs/is missing
4	The two sentences are mostly equivalent, but some unimportant detail differs/missing
5	The two sentences are completely equivalent, as they mean the same thing

3.1.2 DGT-TM Corpora

We use the DGT Translation Memory corpus for our second set of experiments. The DGT-TM is a corpus of aligned sentences in 22 different languages created from the European Union's legislative documents (Acquis Communautaire) (Steinberger et al., 2006). We use the DGT-TM corpora to extract 500 input sentences and retrieve the most similar sentences using the SemEval similarity metric we developed. We extract the proposed French translations in the same way. Our resulting dataset consists of 2500 sentence pairs, their machine translations and their reference translations.

3.1.3 Similarity in Machine Translation Output Dataset

The datasets above face some limitations which prevent our system achieving its full potential. Either these datasets are not designed for similarity (DGT-TM), or they lack a reference translation or another reliable quality rating (SICK).

The limitations of the datasets above lead us to design a new dataset. This dataset consists of a pair of English sentences of variable level of medium to high semantic similarity, and their French machine translations. These sentences are extracted from the FLICKR images dataset used for previous SemEval STS tasks. Each pair has a similarity rating between 4-5, and a French translation created by SMT (using a Moses phrase-based model), with variable levels of translation quality. The quality of the machine translated output is varied by using a more random selection across the n-best list of translations. This ensures that not all translations are of the same quality, as measuring the effect of translation quality on the semantic similarity is an interesting part of our research.

Example 1:

En_1 A group of kids is playing in a yard and an old man is standing in the background

En_2 A group of boys in a yard is playing and a man is standing in the background

Fr_1 Un groupe d'enfants joue dans une cour et un vieil homme est debout dans l'arrière-plan

Fr_2 Un groupe de garçons dans une cour joue et un homme est debout dans l'arrière-plan

Our main objective is to build a dataset where the translations (Fr_1 and Fr_2), are assigned a quality rating and a semantic similarity score.

For this purpose we require two types of human annotations:

1 A quality score for each translation, between 1 and 4, assigned through manual evaluation by a professional translator.

2 A similarity rating for the French sentences produced by the machine translation, achieved through crowd-sourcing.

While this dataset is currently in development, we have already produced a sample set of 1000 sentences to run preliminary experiments on.

3.2 Automatic Evaluation Metric - BLEU

In spite of recent criticisms against it, BLEU remains a widely used metric, especially in evaluating the output of statistical machine translation. We therefore opt to use BLEU as a way to evaluate our system's performance in experiments a human evaluation is not available. As BLEU works best at a document level, we use a sentence level version of BLEU (S-BLEU (Lin and Och, 2004)) to score sentences in cases where no manually annotated score is available. The main difference between BLEU and S-BLEU is that S-BLEU will positively score segments that do not have a higher n-gram matching, unless there is no unigram match.

3.3 MiniExperts Semantic Textual Similarity Tool

SemEval's shared tasks have been particularly interested in semantic similarity, working to fine-tune and perfect these similarity measures, and explore the nature of meaning in language. SemEval2014's Task 1 involves computing how similar two English sentences (Subtask 1a) and whether or not one sentence entails the other (Subtask 2b) (Marelli et al., 2014a). We developed our own Semantic Similarity tool for the SemEval workshop held in 2014 (later expanding on it for the 2015 workshop).

In both workshops we employ a Machine Learning (ML) method which exploits available NLP technology, adding features inspired by deep semantics (such as parsing and paraphrasing) with distributional Similarity Measures, Conceptual Similarity Measures, Semantic Similarity Measures and Corpus Pattern Analysis[1] (CPA). A full description of our system and its performance can be found in our SemEval2015 paper (Béchara et al., 2015).

3.3.1 Features

For these experiments, however, we use a minimalistic version of our tool, restricting it to the 12 most efficient features and sacrificing a small amount of accuracy for speed.

Linguistic Features

We extract a total of 7 linguistic features which using pre-existing language processing tools. In a general sense, these features calculate word overlap between sentences. The aim of these features is to capture token based grammatical similarity between a pair of sentences. To that end, we look at more than just the surface form of these sentences, and extend our features to look at the overlap of parts of speech, lemma, dependency relations and named entities. Overlap is computed using the Jaccard similarity, which is defined in equation 1.

$$Sim(s1, s2) = \frac{|s1 \cap s2|}{|s1 \cup s2|} \tag{1}$$

where $Sim(s1, s2)$ is the Jaccard similarity between sets of words $s1$ and $s2$.

Paraphrasing Feature

The paraphrasing feature aims to detect when a segment is a paraphrase of another segment. To that end, it makes use of the PPDB paraphrase database (Ganitkevitch et al., 2013) to extend each sentence's n-grams with matching n-grams from the database. We then calculate overlap (Jaccard similarity), between these n-grams to get a feature value.

Machine Translation Evaluation Features

In another attempt to capture similarity between two sentences, we turn to BLEU. We extract 3 features using BLEU, based on the sentences' surface form, lemma and parts of speech.

Corpus Pattern Analysis Features

Corpus Pattern Analysis (CPA) is a corpus-driven technique in corpus linguistics and lexicography that associates word meaning with word use by mapping meaning onto specific syntagmatic patterns exhibited by a verb in any type of text (Hanks, 2013). CPA aims at identifying patterns of normal usage (norms), including literal and metaphorical uses, phrasal verbs and idioms, and exploring the way patterns are creatively exploited (exploitations). CPA is currently being used to compile the Pattern Dictionary of English Verbs (PDEV), an online lexical resource that currently covers nearly 1,300 English verbs.

Our final two features make use of the Pattern Dictionary of English Verbs. The first of these features returns 1 when the verb patterns across sentences match, and 0 otherwise. The second feature returns a probability of a PDEV pattern given a specific word. The probability itself is computed over a manually tagged portion of the British National Corpus (BNC).

[1] http://pdev.org.uk

3.3.2 Predicting Semantic Similarity Through Machine Learning

We build a regression model which estimates a continuous semantic similarity score between 0 and 5 for each sentence pair.

We train this system on on a combination of training and trial data provided by the 2012, 2013 and 2014 SemEval tasks. We use these datasets to form a training set of 9750 sentence pairs combining the different domains covered by the STS task: image description (image), news headlines (headlines), student answers paired with reference answers (answers-students), answers to questions posted in stach exchange forums (answers-forum), English discussion forum data exhibiting committed belief (belief). We optimise for the values of C and γ through a grid-search which uses a 5-fold cross-validation method, and all systems use an RBF kernel.

3.3.3 Performance

Our system performed adequately, with our best run achieving a mean Pearson Correlation of 0.7216, as scored by SemEval 2015, ranking 33rd out of 74 systems. Table 2 provides a breakdown of these results.

Table 2: Pearson Correlation - as calculated by SemEval2015

	Pearson Correlation
answers-forums	0.6781
answers-students	0.7304
belief	0.6294
headlines	0.6912
images	0.8109
mean	***0.7216***
rank (out of 74)	***33***

3.4 Translation Model

As we focus on evaluating the output of statistical machine translation, we require machine translated output to test our evaluation systems. To that end, we use a phrase based statistical machine translation system called Moses (Koehn et al., 2007). We build 5-gram language models with Kneser-Ney smoothing trained with SRILM, (Stolcke, 2002), the GIZA++ implementation of IBM word alignment model 4 (Och and Ney, 2003), with refinement and phrase-extraction heuristics as described in (Koehn et al., 2003). We used minimum error rate training (MERT) (Och, 2003) for tuning on the development set.

We trained on 500,000 unique sentences from the Europarl corpus (Koehn, 2005), and then tuned (using MERT) on 1000 different unique sentences. We trained two separate models, one to translate from English to French, and one to translate from French to English.

4 Semantic Textual Similarity as a Tool for Evaluation

The previous research outlined in Section 2 demonstrates the importance of semantic information in the evaluation process. However, the metrics proposed so far all rely heavily on a reference translation. As we have mentioned before, reference translations are not always available. Furthermore, producing reference translations can be labour-intensive and impractical. Quality Estimation addresses this problem by removing with the need for a reference translation. However, the integration of semantic textual similarity into the quality estimation pipeline has not been addressed. The main stumbling block in this process arises due to the monolingual nature of semantic textual similarity and the tools that measure it. Therefore, we propose a system that compares machine translation output to a second sentence, using monolingual STS tools to measure the semantic adequacy of the sentence in relation to the second sentence. The main question we want to answer is whether or not this information, along with a quality score for the second sentence, is enough to assess the translation quality of the machine translation.

The remainder of this section describes our experimental setup and methodology, along with our results and analysis.

4.1 Experimental Setup

Given a sentence pair A_{EN} and B_{EN}, with a translation A_{FR} and B_{FR} respectively, we set out to predict the value of X (an evaluation score for B_{FR}). As we are assuming in this context that neither a quality score nor a reference for B_{FR} exist, we set out to estimate this value based on R (the semantic similarity between A_{EN} and B_{EN}), b_A (the quality score of A_{FR} when compared to a reference translation of A_{FR}) and b_B (the quality score of B_{FR} when compared to a reference translation of A_{FR}). In summary:

$$X = f(R, b_A, b_B)) \tag{2}$$

This set-up is further demonstrated in Figure 1:

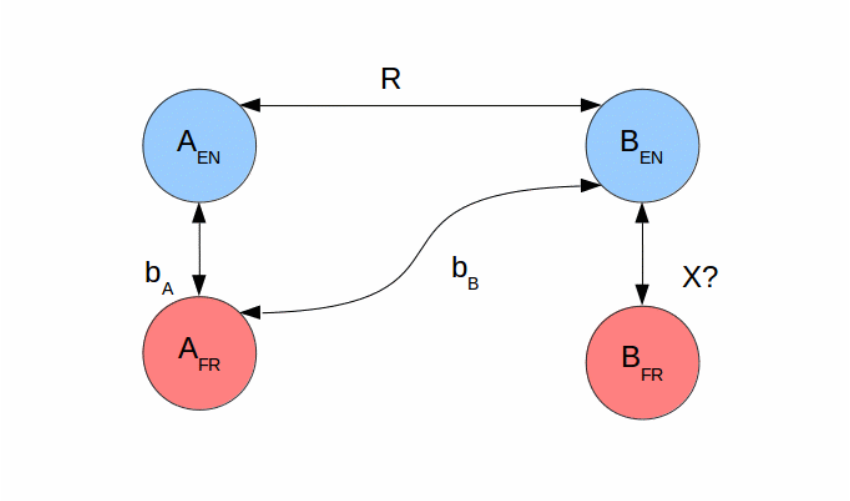


Figure 1: Can we predict the Translation Quality of $B_F R$ (X) as a function of R , b_B and b_A?

4.2 Methodology

Each of our datasets consists of a set of English sentence pairs, A_{EN} and B_{EN}, and a machine translated French sentence for each. All translations were produced using the PB-SMT system Moses, as described in Section 3.4. We use LibSVM[2], a library for SVMs developed by Chang and Lin (2011) in order to predict the quality of the machine translation output.

The first obstacle we face in testing our method is the collection of similar sentences against which to compare and evaluate. During some preliminary experiments, we automatically searched large corpora for sentences that yield high similarity scores. This method proved to be too time-consuming, as it often required scoring thousands of sentences before finding two that were similar. We were able to cut down this processing time by using edit distance as a first-stage filter, returning the 50 sentences with the closest edit distance. However, we were unable to find suitable matches for most sentences. Furthermore, the STS system we designed (See Section 3.3) returned many false-positives, leading to noisy data and unusable results. Therefore, we opt to start from the assumption that we already have access to semantically similar sentences. We use sentences with crowd-sourced similarity ratings in order to produce the best possible results (or oracle score).

4.3 Results

We present three sets of experiments that show positive results. The experiments differ only slightly depending on the data available. The experiments and the subsequent results are detailed in this section.

[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

4.3.1 SICK's Backtranslations

For our first attempt at predicting quality through semantic similarity, we used the SICK corpus detailed in Section 3.1. As SICK already provides us with sentence pairs of variable similarity, it cut out the need to search extensively for similar sentences. Furthermore, the crowd-sourced similarity scores act as a gold standard that eliminates the false positives introduced by the automatic STS tool. This dataset lacks a reliable reference translation to compare against, however.

As we do not have a quality score for our translation, or a French reference translation for this dataset, we opted to use a back-translation (into English) instead of a French translation for these results. A back-translation is a translation of a translated text back into the original language. They are usually used to compare translations with the original text for quality and accuracy, and can help to evaluate equivalence of meaning between the source and target texts. In machine translation contexts, they can be used to create a pseudo-source that can be compared against the original source. He et al (2010) use this back-translation as a feature in quality estimation. They compare the back-translation to the original source using fuzzy match scoring and use the result to estimate the quality of the translation. The intuition here is that the closer the back translation is to the original source, the better the translation is in the first place. Following this idea, we use the S-BLEU scores of the back-translations as stand-ins for the machine translation quality scores. This means we translated the 5,000 sentences into French and then translated the French output back into English, using both models described in Section 3.4. We then compared the resulting MT output to the original English sentences to produce the sentence level BLEU (S-BLEU) (Papineni et al., 2002) scores. We chose to use S-BLEU in these experiments, as it is a popular metric designed for sentence-level evaluation. Despite the short-comings of using back-translations, the advantage of using this dataset lies in the crowd-sourced similarity ratings, which provide us with a gold standard for semantic textual similarity ratings. These ratings are unique to datasets produced by SemEval.

We build a support vector regressor using the STS rating and the BLEU scores of both Sentence A and Sentence B, using the original Sentence A as a reference for both sentences. We attempt to predict the basic S-BLEU score of Sentence B (using the original Sentence B as a reference). Results on this dataset are promising, producing a Mean Absolute Error of 0.193; better than the mean baseline (which is calculated using the mean of the scores in the training set in all cases in the test set).

Our results are summarised in Table 3.

	Mean Baseline	STS (3)
MAE	0.216	0.193

Table 3: Predicting the S-BLEU scores for SICK's Backtranslations - Mean Absolute Error

However, the use of back-translations in lieu of actual translations is a potential problem. Back-translations are not always a good indication of the quality of the original machine translation. Criticisms of back-translations in machine translated output have shown that the quality of a back-translation does not always match up with the quality of the first-tier translation (Somers, 2005). Somers (2005) shows that a garbled sentence might be translated back to match the source, while a good translation could become distorted during back translation. We attempt to address this short-coming in the rest of this section by turning to different corpora and even designing our own.

4.3.2 DGT-TM

We apply the same technique we used on the SICK corpus in order to predict the quality of the semantically similar sentences extracted from the DGT-TM. We randomly select 500 sentences for testing and use the remainder 2000 to train our model.

We compare our results to both the QuEst baseline (17 baseline features offered as a baseline for QuEst) (Specia et al., 2013) and the mean baseline (using the mean rating as a projected score for every sentence). The lowest error rate is observed for the system that combined our STS-based features with QuEst's baseline features (Combined (20)). Even the 3 STS features on their own outperformed QuEst's

baseline features for this particular dataset. These results show that despite the short-comings of the STS-tool, our method can prove useful in a context where semantically similar sentences are accessible.

Table 4 lays out these results.

	Mean Baseline	QuEst Baseline (17)	STS (3)	Combined (20)
MAE	0.16	0.12	0.108	0.09

Table 4: Predicting the S-BLEU scores for DGT-TM - Mean Absolute Error

4.3.3 Similarity in Machine Translation Output Dataset

While BLEU is widely used to evaluate MT systems today, it still has severe short-comings when it comes to correlating with human judgement. This sheds some doubt on the previous experiments, which rely heavily on sentence-level (S-BLEU) scores to evaluate our system. Therefore, we chose to use human judgements to evaluate against. We run preliminary experiments mirroring those run on the DGT-TM corpus on the new dataset that we designed. We use 200 randomly chosen sentences as a test set, and use the remaining 800 sentences to train our machine learning system. As the quality scores for the dataset we designed are discrete (1-5) as opposed to the continuous S-BLEU scores used in previous experiments, we use a SVM classifier rather than a regressor. Our results show that the addition of the STS-related features can improve our predictions marginally over those of QuEst's baseline features. However, further investigation with the full dataset would be required before any concrete conclusions can be drawn.

	QuEst Baseline	Baseline + STS
Accuracy	40%	45%

Table 5: Classification Accuracy for New Dataset

While these results are promising, an 800 sentence training set is severely limiting, and we cannot forge concrete conclusions without further investigation.

5 Conclusions

This paper presents our investigation into the use of semantic textual similarity in reference-free machine translation evaluation. We present a series of experiments using different corpora and even design our own dataset specifically for this task. We designed a machine learning system using SVM to predict the quality of a machine translated sentence based on its source similarity to another sentence's source, and the other sentence's MT quality. Preliminary experiments faced obstacles in identifying semantically similar sentences to use for evaluation. However, when we used datasets specifically designed for similarity, we were able to achieve positive results, improving on QuEst's baseline in a quality estimation context by integrating semantic textual similarity as a feature.

Our results are optimistic, despite the use of S-Bleu to evaluate our system. Criticisms of BLEU and n-gram matching metrics in general are addressed by Callison-burch et al. (2008), who show that BLEU fails to correlate to (and even contradicts) human judgement. More importantly, BLEU itself does not measure meaning preservation. Therefore, to evaluate our system more thoroughly, we would need to expand the dataset described in Section 3, which relies on human judgements for evaluation. Furthermore, our features rely on the existence of semantically similar sentences against which we can compare our translations. These sentences are not always readily available, and searching large corpora for similar sentences can be computationally costly and time-consuming. However, this approach can be quite useful in settings where we wish to evaluate sentences within a very specific domain, where highly similar documents and sentences are available.

In the future, we plan a more thorough investigation of the results and an analysis of our system once the dataset we are building is complete. Furthermore, we plan to look at semantic similarity between

a pair of machine translated sentences and determine how well semantic similarity is preserved after machine translation. We plan to investigate ways to predict the similarity of the machine translated sentences based on the similarity of the source sentences, and the quality of the MT output.

Acknowledgements

Hanna Béchara is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471. This work is made possible with the support and supervision of Dr Constantin Orăsan and Dr Lucia Specia.

References

Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. 2015. MiniExperts: An SVM approach for Measuring Semantic Textual Similarity. In *9^{th} Int. Workshop on Semantic Evaluation*, SemEval'15, pages 96–101, Denver, Colorado, June. ACL.

P. Brown, J. Pietra, S. D. Pietra, F. Jelinek, R. Mercer, , and P. Roossin. 1990. A Statistical Approach to Machine Translation. In *Computational Linguistics*, pages 16:79–85.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation (WMT)*, pages 70–106.

Julio Castillo and Paula Estrella. 2012. Semantic textual similarity for mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 52–58. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Jesús Giménez and Lluís Màrquez. 2007. Linguistic features for automatic evaluation of heterogenous mt systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264. Association for Computational Linguistics.

P. Hanks. 2013. Lexical Analysis: Norms and Exploitations.

Y. He, Y. Ma, J. van Genabith, and A. Way. 2010. Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 622–630.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 177–180.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chi-kiu Lo and Dekai Wu. 2011. Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 220–229. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *8^{th} Int. Workshop on Semantic Evaluation*, SemEval-2014, Dublin, Ireland.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A sick cure for the evaluation of compositional distributional semantic models. In *LREC'14*, Reykjavik, Iceland.

Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. In *Proceedings of the Association for Computer Linguistics (ACL)*, pages 29(1):19–51.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 311–318.

Harold Somers. 2005. Round-trip translation: What is it good for. In *Proceedings of the Australasian Language Technology Workshop*, pages 127–133.

L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. 2009a. Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Meeting of the European Association for Machine Translation (EAMT-2009)*, pages 28–35.

Lucia Specia, Marco Turchi, Zhuoran Wang, John Shawe-Taylor, and Craig Saunders. 2009b. Improving the confidence of machine translation quality estimates.

L. Specia, D. Raj, and M. Turchi. 2010. Machine Translation Evaluation versus Quality Estimation. In *Machine Translation Volume 24, Issue 1*, pages 39–50.

L. Specia, K. Shah, J. Guilherme, C. de Souza, and T. Cohn. 2013. QuEst - A translation quality estimation framework. In *Proceedings of the Association for Computational Linguistics (ACL), Demonstrations*.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, and Dan Tufis. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC–2006*, pages 2142–2147.

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904.