Journal of Intelligent & Fuzzy Systems 29 (2015) 2769–2777
DOI:10.3233/IFS-151981
IOS Press

Domain mining for machine translation

Junfei Guo[a,b], Juan Liu[a,*], Qi Han[c], Xianlong Chen[d] and Yi Zhao[a]
[a]*School of Computer, Wuhan University, Wuhan, China*
[b]*Institute for Natural Language Processing, University of Stuttgart, Stuttgart, Germany*
[c]*Institute for Visualization and Interactive Systems, University of Stuttgart, Stuttgart, Germany*
[d]*City College of Dongguan University of Technology, Dongguan, China*

Abstract. Massive amounts of data for data mining consist of natural language data. A challenge in natural language is to translate the data into a particular language. Machine translation can do the translation automatically. However, the models trained on data from a domain tend to perform poorly for different domains. One way to resolve this issue is to train domain adaptation translation and language models. In this work, we use visualizations to analyze the similarities of domains and explore domain detection methods by using text clustering and domain language models to discover the domain of the test data. Furthermore, we present domain adaptation language models based on tunable discounting mechanism and domain interpolation. A cross-domain evaluation of the language models is performed based on perplexity, in which considerable improvements are obtained. The performance of the domain adaptation models are also evaluated in Chinese-to-English machine translation tasks. The experimental BLEU scores indicate that the domain adaptation system significantly outperforms the baseline especially in domain adaptation scenarios.

Keywords: Text clustering, domain detection, domain adaptation, language models, machine translation

1. Introduction

Data mining [13] is the process of extracting interesting patterns or knowledge from huge amounts of data. The majority of real-world data is in the form of natural language text. One of the grand challenges is to translate the text from one language into another language. Statistical machine translation (MT) [20] is proposed to automatically generate translation results by statistical models. The models are trained from parallel and monolingual corpora. Language models (LM) [4] help machine translation system to discriminate alternative target hypothesis by assigning higher probability to a more fluent sentence.

An issue in MT is that the models trained on data from a particular domain will perform poorly when translating texts from cross-domain data. On the one hand, it is not easy to obtain bilingual corpora for specific domains. Even huge in-domain training data is typically insufficient to combat cross-domain data sparseness. Facing the out-of domain MT tasks, limited models are explored for automatically discovering domains in MT corpora. On the other hand, due to the lack of domain data, language models typically do not make use of domain knowledge in the target data. In general, the accurate estimation of cross-domain n-grams is difficult to achieve by using only knowledge about in-domain n-grams.

In this work, we propose novel methods to detect the domain of the test data based on text clustering and domain language models. These methods are used to classify the test data, choose the domain data for the target set. The domain data will be used for language model adaptation or interpolation. Furthermore, we present domain adaptation language models in machine translation. The domain adaptation LMs can easily be adapted to different domains via a parameter tuning step. We train the principal model with a large amount of

*Corresponding author. Juan Liu, School of Computer, Wuhan University, Wuhan, China. Tel./Fax: +86 27 6877 5711; E-mail: liujuan@whu.edu.cn.

ISSN 1064-1246/15/$35.00

background training data and then use the domain data to adjust its discount parameters. In addition, we linear interpolate the LM with a domain language model, in which the interpolation weights are optimized to minimize the perplexity [8] on a domain development set. Finally, we implement the novel models and integrated them into a standard machine translation pipeline.

In our experiments, the domains of the test data can be classified and discovered by our domain detection methods. Our domain adaptation LM was compared with the well-known popular modified KNESER-NEY models [24] that are implemented in SRILM [3]. For in-domain test data, our model has lower perplexity scores than the Kneser-Ney model. For cross-domain, our LM achieves significant better perplexity than the competitor. Our model is evaluated on a Chinese-to-English translation task by using both the phrase-based [23] and the hierarchical phrase-based [5] translation model of Moses [21]. The obtained translation results indicate that our model significantly outperforms the baselines. The improvement is particularly large in domain adaptation scenarios.

2. Related work

Domain adaptation for machine translation has been the focus of several researchers in recent years. Amittai et al. [1] explored domain adaptation by select sentences from a large general domain parallel corpus that are most relevant to the target domain. Xu et al. [14] used the combination of feature weights and language model adaptation, to distinguish multiple domains. Sennrich [22] investigated translation model perplexity minimization as a method to set model weights in mixture modelling. Hasler et al. [7] investigated that the combining domain and topic adaptation approaches can be beneficial and that topic representations can be used to predict the domain of a test document.

A large number of studies have proposed and investigated language models for machine translation. The most popular smoothing method in statistical machine translation is the Kneser-Ney (KN) models [24], which are implemented in LM toolkits such as SRILM [3] and KenLM [15]. A polynomial discounting mechanism [10] was proposed by Schütze in the POLO model, which is a class-based LM that interpolates additional classes.

In this work, we investigate the domain adaptation language models based on tunable discounting parameters and domain interpolation. We also address the domain detection methods based on text clustering and domain language models.

3. Domain detection

In this section, we use two domain detection techniques to discover domains automatically. If the techniques can detect the domain of the test data, then the MT system can choose the optimal domain adaptation models to improve the results on translation tasks.

To evaluate the domain discrimination methods, we perform experiments by using a standard MT tuning and test set from different domains as experimental data to classify the domains. The data are Chinese-English parallel data covering the following 6 domains: Biology (Bio), Food (Food), Semiconductor (Semi), Social Media (Social), Newswire (News) and Web News (Web). For each domain, the data has two parts including tuning set and test set. In the following sections, two methods will be used to detect the domains of the test sets.

3.1. Topic clustering for domain detection

In the first method, we use Latent Dirichlet allocation (LDA) [6] for the documents clustering and cosine similarity to perform the domain detection task.

We put all the tuning sets and test sets together as the LDA training set. The training data are normalized and all the stop words are filtered out. We set the number of topics to 6 and do the document clustering on the training set. After fitting on the documents, LDA provides inferred topics, probability distribution over words and document probability distributions over topics. The sentences in the training set are then labeled with domain name for the domain detection purpose.

The document probability distributions over topics are used for text domain matching. The attributes of the training text vectors and test text vectors are the topic distribution vectors of the documents. Cosine similarity is used to measure the similarity between two vectors of documents. The resulting similarity ranges from 0 to 1 and in-between values indicating intermediate similarity of the text domains.

The similarities between all the training set and test set are illustrated in Table 1. In the first line of Table 1, we can see that the similarities between the Bio tuning set and other test sets range from 0.9999 to 0.0922. The highest similarity (0.9999) points to the Bio test set. The first three lines show that the similarities between

Table 1
Cosine similarities of different domain

Training Set	Test Set					
	Bio	Semi	Food	Social	Web	News
Bio	**0.9999**	0.1022	0.0922	0.1288	0.1293	0.1255
Semi	0.1012	**0.9999**	0.0763	0.1338	0.1131	0.1075
Food	0.0948	0.0796	**0.9998**	0.4443	0.3243	0.2688
Social	0.1410	0.1311	0.4376	**0.9995**	0.8383	0.6748
Web	0.1257	0.1287	0.3355	0.8633	**0.9996**	0.9519
News	0.1328	0.1093	0.2358	0.6500	0.9503	**0.9739**

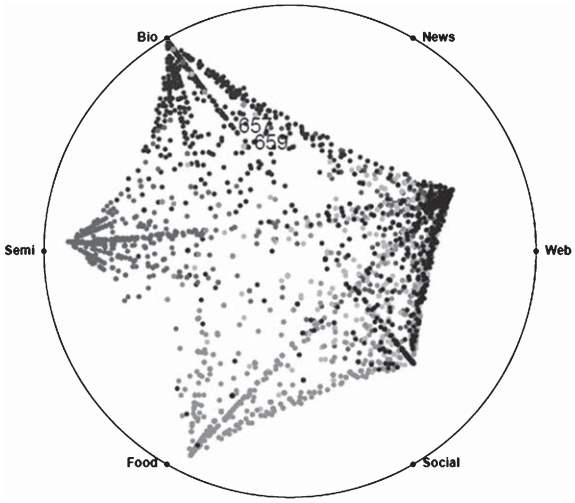

Fig. 1. Sentences clustering.

the same domains are close to 1 (Semi 0.9999 and Food 0.9998), which indicate that the text clustering method can detect the domain well especially when the domains are quite different from each other (such as Bio, Semi and Food).

However, in the last three lines of the table, we can see that the similarities between the Social media, News and Web news are close to each other, which is also true in the fifth line, where the similarities between the Web tuning data and the test set from Social, News and Web news are 0.8633, 0.9519 and 0.9996. It is not easy to discriminate the Newswire data and the Web news data, since the similarities between Newswire and Web news are close to 0.97. As we know, the two domains are very close to each other.

We also use visualizations to show the similarities of the sentence. We created a new interactive visualization to enable a deeper analysis of the possibles errors of this method. Figure 1 is a visualization based on Radvis [17]. Firstly, it depicts the similarities between the sentences in the test set and the inferred topics. The dots represents the sentences and the color of the dots corresponds to the gold stand domains. In this figure we can see that most of the sentences from domains "Bio", "Semi", and "Food" are easy to recognize. In contrast, many sentences from the domains "News", "Social", and "Web" are difficult to differentiate, because they also similar to each other.

3.2. LMs for domain detection

In the second method, domain LMs were used for the domain detection. We trained LMs on the same tuning sets covering 6 domains. The LMs on the tuning set were used to calculate perplexities on the test data. We can use the perplexities matrix to detect which domain the test set belongs to. The lowest perplexity point to the test set with closed domain. In this way, we consider the tuning set and the test set are from the same domain. SRILM are used to train 5-gram language models on Chinese domain data.

All the perplexities for the different test sets are shown in Table 2. The first line of this table show the results of the LM trained on the Biology tuning set. The LM was used to calculate perplexities for the test sets from the different 6 domains. It is shown that the perplexity for the test set of Biology data is 340.6. which is lower than other perplexities (from 2403.8 to 6155.5). In other lines of this table, we can also see that the lowest perplexities of the test data always from the same domain LMs. In the fifth line of this table, we can observe that the LMs is from Web News data, the lowest perplexity comes from Web data (852.8) and the second lowest perplexity from news domain (1428.8). The two perplexities are closer than the others since the two domains are more similar than other domains. This result confirms that the lowest perplexity matches the most similar domains. The experiment results indicate that the domain LMs can effectively detect the domain of the test data.

Comparing the results of the last line in Table 1 and Table 2, we can see that the domain LM method can

Table 2
Perplexities for different domain test set

Training LMs	Test Set					
	Bio	Semi	Food	Social	Web	News
Bio LM	**340.6**	2403.8	5148.1	6155.5	5360.2	5100.3
Semi LM	4023.0	**305.4**	8807.5	6606.3	7944.5	7647.3
Food LM	4869.9	4649.5	**633.3**	1771.2	2345.6	2365.6
Social LM	5726.9	4516.1	2398.7	**736.5**	2014.0	1818.8
Web LM	4108.1	3581.9	1925.5	1443.2	**852.8**	1428.8
News LM	5966.4	5933.8	3015.5	1920.0	2072.4	**995.5**

easily distinguish between the web news and the newswire data by large perplexities difference (995.5 and 2072.4). By analyzing the topic (e.g., business, biology) and genre (e.g., newswire, web-blog), we can see that the differences across topics are limited to domain. The text clustering based method could not classify the text such as newswire and web news since they share similar topics. Our LM methods can discriminate well the genre specifics of the two texts.

In both of these two methods, we only use the source language data (Chinese in our experiment) to detect which domain the test set belongs. While we got the domain of the source language, we use the parallel data from the target language (English in our experiment) for our domain adaptation LM optimization. We can use the target language data to optimize the parameters of our domain adaptation language models or do the domain interpolation for MT system.

4. Domain adaptation language models

The KN model [24] smooths the LM by a discounts $D(c)$ which are trained on the training corpora. We proposed a simpler n-gram model, the polynomial-discount domain-adaptation LM (DA) [11] in our earlier work by using polynomial discounting parameters, which are trained on a validation data. We improve the discounting function in our recent work [12] with more parameters and replace the KN discount $D(c)$ by the discounting function:

$$E'(c) = D(c) + \begin{cases} t_c & \text{if } c \leq 3 \\ \rho \cdot c^{\gamma} & \text{otherwise} \end{cases}$$

There are 5 parameters in this model, t_c representing three constant parameters t_1, t_2 and t_3, respectively. When the n-grams are low-frequency events ($c \leq 3$), we only tune the KN discount $D(c)$ by adding a parameter t_c. In this way we can use the advantage of KN discount and tune the discounts to new domain. However, when the n-grams are high-frequency events ($c > 3$), we use the sum of the modified KN discount $D(c)$ and the polynomial discount function with parameters ρ and γ. We use this polynomial discount function to replace the t_c to adapt the high-frequency events in this case, because KN discount still keeps the discount $D(c)$ as $D(3)$ even $c > 3$. In this way, we use the advantages of KN and polynomial discount and avoid the disadvantages of them. All the parameters added to the modified KN discount function $D(c)$ need to be optimized on the target data. We call this tunable and polynomial discounting KN Model (TPKN) defined as:

$$P_{tpkn}(w_i|w_{i-n+1}^{i-1}) = \begin{cases} \frac{c(w_{i-n+1}^{i}) - E'(c(w_{i-n+1}^{i}))}{\sum_{w_i} c(w_{i-n+1}^{i})} \\ \quad \text{if } c(w_{i-n+1}^{i}) > 0 \\ \\ \beta(w_{i-n+1}^{i}) P_{tpkn}(w_i|w_{i-n+2}) \\ \quad \text{if } c(w_{i-n+1}^{i}) = 0 \end{cases} \tag{1}$$

This LM model is a simple, recursive model with tunable discounting parameters and polynomial discounts. We use a simple back-off scheme for smoothing. Heuristic grid search is used to find the optimal parameters.

5. MT evaluation experiments

5.1. Setup

To evaluate our approaches, we apply the domain detection methods and domain adaptation LMs on MT system in the following sections.

In the domain detection step, we use LDA and domain LMs to work on text classification. Cosine similarities and perplexity are used to discover the text domain. After the domain detection step, additional data from the same domain will be chosen as the development data.

These data will be used to optimize the parameters of LMs or for domain interpolation. As already mentioned, we use a tuning step with the development data to set those discounting parameters in TPKN model. During heuristic grid search for the optimal discounting parameters, we explore the space with step size 0.01. we have modified the SRILM implementation of discounting in order to implement our TPKN model. We use SRILM modified KN model to generate 5-gram LMs on English data.

For the MT evaluation experiments, we used the statistical machine translation framework MOSES [21]. GIZA++ [9], which is part of MOSES, with the alignment "grow-diag-final-and" heuristic [18] was used to obtain the word alignments. The translation models were trained with parallel sentences after length-ratio filtering. We use BLEU [16] score to evaluate MT results for the test sets.

5.2. Corpora

A special release of the sentence-aligned MultiUN corpus [2] was used as training data, which is a multilingual parallel corpus extracted from official documents published by the United Nations from 2000 to 2009. This corpus is available in all 6 official languages of the UN and contains roughly 300 million words per language. The Chinese-English part was made available in August 2011 for IWSLT 2011, of which, we only use 2 million parallel Chinese-English sentences as MT training data. 8.8 million English sentences were used as general LM training data.

The evaluation is based mostly on NIST data. The NIST data consists of newswire documents, human transcriptions of broadcast news documents collected by the LDC. The NIST documents are quite different (in style and language) from the contract documents of MultiUN. Thus, we consider the MultiUN and the NIST data as data from different domains. NIST 2002, 2004, 2006 was used as tuning data and NIST 2005 as test data. The NIST 2005 data are exactly the same test data from newswire domain as we used in the domain detection steps. After we got the domain of the test language, we can apply the domain data for the MT system.

In addition, we used more than 15,000 lines Newswire domain data from the NIST 2009, -10, -11, -12, -13, -14, for the domain interpolation LM. Obviously, the domain of the additional NIST data is the same as our target test data. With the additional domain data, we interpolated our LM for the training set with an additional LM for the domain data.

6. Results and discussion

The outcomes of our experiments are listed in Tables 3–5. Let us discuss the individual experiments one-by-one.

6.1. Language model perplexity experiments

The first experiment investigates the perplexity performance of the different LMs on cross-domain data. For comparison, we also report the results obtained on in-domain data. We train the language models on the training data and evaluate the LMs perplexities on different domain validation and test sets.

For the in-domain experiment, the training set, the validation set, and the test set are from the English part of MultiUN. The training set contains more than 2 mil-

Table 3
LM perplexities on in-domain and cross-domain data

Language model	In-domain			Cross-domain		
	Val	Tst	Size	Val	Tst	Size
KN	54.55	60.64	584.07 MB	289.92	312.30	2.20 GB
KenLM	**49.14**	**52.13**	2.80 GB	296.51	317.94	9.40 GB
DA	52.39	58.33	583.80 MB	271.20	286.53	2.19 GB
TPKN	52.50	58.34	**583.03 MB**	**270.24**	**284.99**	**2.19 GB**

Table 4
BLEU scores for in-domain and cross-domain MT

Language model	In-domain		Cross-domain	
	PBMT	HPBMT	PBMT	HPBMT
KN	31.32	34.30	20.05	20.64
DA	31.34	34.78	20.35	**20.95**
TPKN	**31.43**	**34.87**	**20.42***	20.93*

Table 5
BLEU scores for cross-domain MT

Model	PBMT	HPBMT
KN	20.05	20.64
DA	20.35	20.95
TPKN	20.42	20.93
KN+domain	20.50	20.87
DA+domain	20.51	20.96
TPKN+domain	**20.60***	**20.98***

lion lines and both the validation and the test set have roughly 2,000 lines. All perplexity values are reported for the validation (Val) and the test (Tst) set.

For the cross-domain experiment we use the English part of the MultiUN corpus [2] as training corpus (8.8 million sentences). As mentioned, we also use the English part of the NIST 2002, 04, 06 as validation set and NIST 2005 as the test set.

In the experiment, we compare the standard modified KN language models provided by the toolkits SRILM and KenLM. KenLM uses a no-pruning strategy, which it compensates for with its high efficiency allowing it to handle the resulting large models. Since our models and the KN models in SRILM rely on pruning to reduce the size of the models, we select the modified KN model implemented in SRILM (KN) as baseline. We also compare the TPKN model with the recently related work polynomial-discount domain-adaptation LM (DA) [11].

Table 3 shows the performance of all LMs. Our baseline is the modified KN model offered by SRILM. In the in-domain case, we observe improvements in perplexity (from 54.55 to 52.50) compared to our baseline on the validation set. Similarly, we observe a slight improvement in perplexity from 60.64 to 58.34 on the test set. When we compare the TPKN model with the DA model (52.39 on Val, 58.33 on Tst), we observe slight improvement. These results suggest that our TPKN model slightly improves LM performance even on in-domain data. KenLM obtain the lowest perplexities (49.14 on Val, 52.13 on Tst), however, the model produced by KenLM is much larger (2.80 GB) than other models in SRILM (< 600 MB). We currently employ the same pruning strategy as SRILM, so our models are small (583.03 MB) compared to models of KenLM and have essentially the same size as the standard SRILM KN models. It remains to be seen whether the reported advantages can also be obtained using a no-pruning strategy as in KenLM together with our model.

When we compare in-domain and cross-domain performance, we observe that the perplexities increase from roughly 50 to roughly 280, which shows that the NIST data is rather different from the training data. The results in Table 3 indicate that our model can achieve considerable perplexity improvements for cross-domain data. The perplexity drops from 289.92 (KN) to 270.24 (TPKN). The results are mirrored on the test set, where we observe an improvement from 312.30 (KN) to 284.99 (TPKN). The scores on the validation and test set are similar as expected because the various NIST data sets are comparable. Compared with the DA model (271.20 on Val, 286.53 on Tst), our model also achieve slightly but considerable improvement. The KenLM performs worst on cross-domain data (296.51 on Val, 317.94 on Tst) with the largest size (9.40 GB) compared with SRILM models (2.19 GB). Since our work focus on the cross-domain scenario, we do not compare the result of KenLM in the following experiments. Overall, our model outperforms the modified KN model in SRILM. The tuning of the discount parameters for our model on the cross-domain validation set helps our model adapt well to new data.

In summary, the perplexity experiments show that our model does not suffer worse performance than the competitor on in-domain data, but offers benefits on cross-domain data. This suggests that we can safely use our model both on in-domain and cross-domain data.

6.2. *Cross-domain machine translation experiments*

Following our experiments with perplexity, we applied the domain adaptation LMs to a Chinese-to-English translation task using both hierarchical phrase-based (HPBMT) [5] and phrase-based (PBMT) [23] translation models of Moses [21]. The various obtained translation systems are evaluated automatically with BLEU [16].

For the machine translation experiments, roughly 2 million Chinese-English parallel sentence pairs from the MultiUN corpus were used as our training data. The same validation and test sets were used but the Chinese part of those data are now also used. In this experiment, the validation set is considered as tuning set in the Moses framework. We measured the overall translation quality with the help of BLEU [16] which was computed on tokenized and lowercased data. The obtained BLEU scores are shown in Table 4. The results are from phrase-based machine translation model (PBMT) and hierarchical phrase-based model (HPBMT). Stars

indicate significant BLEU score improvements over the baseline (at confidence level 95%).

In the in-domain MT, all the BLEU score are higher than 30, which confirms that MT models perform well in the translation of text from the same domain data. Our LM outperformed the KN baseline and the DA model in both scenarios (phrase-based and hierarchical translation model) in which we observed slight improvements from 31.32 (KN), 31.34 (DA) to 31.43 (TPKN) with phrase-based translation model and from 34.30 (KN), 34.78 (DA) to 34.87 (TPKN) with hierarchical translation model.

By comparing, cross-domain performance with in-domain, the BLEU scores decrease from roughly 30 to roughly 20. Given that the training set and test are from the same corpora (MultiUN), the MT system can achieve a high performance in this in-domain experiment; In the meantime, the difference between the training set (MultiUN) and the test set (NIST) should be the reason why the BLEU scores drop. This can be explained by the observation that even huge in-domain training data is typically insufficient to combat cross-domain data sparseness. The overall translation quality allows us to study how to improve the translation performance even if we do not have enough in-domain data.

Cross-domain results show that our model retains a benefit in both scenarios (phrase-based and hierarchical translation model). We observe significant improvements from 20.05 (KN) to 20.42 (TPKN) with phrase-based translation model and from 20.64 (KN), to 20.93 (TPKN) with the hierarchical translation model. This improvement is statistically significant at confidence $p < 5\%$, which we calculated on pairwise bootstrap resampling methods [19]. The DA model performs similar (20.95) as the TPKN model in In hierarchical translation model. However, the TPKN models outperforms the DA (20.35) model in phrase-based translation model.

Overall, the results indicate that our model outperforms KN in terms of both perplexity and translation tasks, especially in cross-domain scenarios.

6.3. *Domain-interpolated machine translation experiments*

Another solution to the domain adaptation issue is to interpolate the LM with an additional LM for the target domain. These mixture models weights are tuned on the development set. This approach utilizes both types of data independently, which can be beneficial. However, to estimate an LM for the target domain, a substantial training data in the target domain is required.

In this experiment, we used an additional small target training set from the NIST 2009, −10, −11, −12, −13, −14. The domains of the NIST data is the same as our target test data. Consequently, the validation set and the test set are still the same as the previous experiment.

Firstly, we trained a big LM on large training data (MuliUN) as previous experiments. In the meantime, we trained another small LM on the small target domain data (NIST2009-14). We interpolated the big LM with the small LM trained on NIST data. The language model mixture weights were optimized to minimize the perplexity measured on the development data. Finally, we could evaluate the test data by the mixed language model with the optimal interpolation weight. Since the target domain training set is small, we simply used the KN model to train the LMs on the domain data. All the models were interpolated by this KN model. Finally, we implemented the LMs to the same Moses setup as in the previous experiment for the machine translation evaluation. This experiment investigated the performance of the different interpolated LMs on cross-domain machine translation.

The results are shown in Table 5. In order to compare our domain-interpolated LMs with the previous raw language models, we put the results of previous raw models in the first three lines of this table. The BLUE scores show that the domain interpolated LMs outperform the raw models without domain data. KN interpolated model obtain a BLEU score of 20.50 (phrase-based translation model), which is a gain of 0.45 BLEU points over the baseline (20.05). The KN interpolated model also achieve a BLEU score from 20.87 to 20.64 for hierarchical translation model. The TPKN domain-based model is also better than the TPKN model without domain interpolation, from 20.60 to 20.42 (phrase-based translation model) and from 20.98 to 20.93 (hierarchical translation model). With the additional domain data, the TPKN models still outperform the DA model (20.51 and 20.96). This improvement is not statistically significant. However, the domain-interpolated TPKN model outperforms all other systems including the KN interpolated domain model.

Comparing the results of the domain-interpolated KN model with the raw TPKN model, we can see that the TPKN model performs as well as the domain-interpolated KN model. The TPKN model even achieves a BLEU score of 20.93 which is higher than the domain-interpolated KN model (20.87). This

improvement is not significant but considerable for cross-domain test set, because the raw TPKN model does not use any domain data for LM interpolation.

Overall, these results suggest that the small target domain-based interpolation LM slightly improves LM performance on domain data. Our TPKN model with interpolation can still outperform KN model. Compared with the domain-interpolated models, the raw TPKN models achieved considerable improvement even if they do not have additional data.

7. Conclusion

In this paper, we utilized visualizations to show the similarities of different domains and addressed methods to detect the domain of the test data based on text clustering and domain language models. These methods were used to classify the test data and to discover the domain of the target set. With the domain data, we proposed a domain adaptation language model TPKN which can easily be tuned to new domains and is thus ideally suited for domain adaptation. We also interpolate this language model with a small specific language model trained on the domain we discovered. Perplexity shows that our models outperform the baseline model especially in domain adaptation scenarios. We implemented our models in the Moses statistical machine translation framework, in which we also observed significant BLEU score improvements.

Future research work will be to improve our work including a more efficient algorithm for domain clustering. We would also like to apply our model to more different domain data.

Acknowledgments

Junfei Guo acknowledges the support by Chinese Scholarship Council during his PhD studies at the University of Stuttgart. We also gratefully acknowledge the financial support by Supported by National Natural Science Foundation of China (61202031), EU Project iPatDoc grant 606163, Young Teachers Development Fund of City College of Dongguan University of Technology. All authors want to sincerely thank the colleagues and anonymous reviewers for their helpful comments, especially Ina Rösiger, Max Kisselew, Jason Utt and Christian Scheible at the University of Stuttgart.

References

[1] A. Amittai, X. He and J. Gao, Domain adaptation via pseudo in-domain data selection, *In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, Edinburgh, UK, 2011, pp. 355–362.

[2] A. Eisele and Y. Chen, Multiun: A multilingual corpus from united nation documents, *In Proceedings of the Seventh conference on International Language Resources and Evaluation*, Valletta, Malta, 2010, pp. 2868–2872.

[3] A. Stolcke, Srilm - an extensible language modeling toolkit, *In Proceedings International Conference on Spoken Language Processing, Denver*, Colorado, USA, 2002, pp. 901–904.

[4] C.D. Manning and H. Schütze, *Foundations of statistical natural language processing*, MIT Press, Cambridge, Massachusetts, 1999.

[5] D. Chiang, A hierarchical phrase-based model for statistical machine translation, *In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics ACL '05*, Stroudsburg, PA, USA, 2005, pp. 263–270.

[6] D.M. Blei, Andrew Y. Ng and M. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* **3** (2003), 993–1022.

[7] E. Hasler, B. Haddow and P. Koehn, Combining Domain and Topic Adaptation for SMT, *In Proceeding of The eleventh biennial conference of the Association for Machine Translation in the Americas (AMTA)*, Vancouver, BC, Canada, 2014, pp. 139–151.

[8] F. Jelinek, R.L. Mercer, L.R. Bahl and J.K. Baker, Perplexity–a measure of the difficulty of speech recognition tasks, *Journal of the Acoustical Society of America* **62** (1977).

[9] F. Och and H. Ney, A systematic comparison of various statistical alignment models, *Comput Linguist* **29** (2003), 19–51.

[10] H. Schütze, Integrating history-length interpolation and classes in language modeling, *In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, 2011, pp. 1516–1525.

[11] J. Guo, J. Liu, Q. Han and A. Maletti, A tunable language model for statistical machine translation, *In Proceeding of The Eleventh Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, Vancouver, BC, Canada, 2014, pp. 357–368.

[12] J. Guo, J. Liu, X. Chen, Q. Han and K. Zhou, Tunable Discounting Mechanisms for Language Modeling, *In Proceeding of Intelligence Science and Big Data Engineering(IScIDE)*, Suzhou, China, 2015.

[13] J. Han, M. Kamber and P. Jian, *Data Mining: Concepts and Techniques*. 3rd edn, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[14] J. Xu, Y. Deng, Y. Gao and H. Ney, Domain Dependent Statistical Machine Translation, *In Proceedings of Machine Translation Summit*, Copenhagen, Denmark, 2007, pp. 515–520.

[15] K. Heafield, I. Pouzyrevsky, J.H. Clark and P. Koehn, Scalable modified Kneser-Ney language model estimation, *In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 690–696.

[16] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, *In Proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, Philadelphia, Pennsylvania, 2002, pp. 311–318.

[17] P. Hoffman, G. Grinstein, K. Marx, I. Grosse and E. Stanley, DNA visual and analytic data mining, *In Proceedings of the 8th Conference on Visualization '97, Phoenix*, Arizona, USA, 1997, pp. 437–441.

[18] P. Koehn, A. Axelrod, R.B. Mayne, C. Callison-burch, M. Osborne and D. Talbot, Edinburgh system description for the 2005 iwslt speech translation evaluation, *In Proc International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, USA, 2005, pp. 68–75.

[19] P. Koehn, Statistical significance tests for machine translation evaluation, *In Proc EMNLP*, Barcelona, Spain, 2004, pp. 388–395.

[20] P. Koehn, *Statistical Machine Translation*. 1st edn, Cambridge University Press, New York, NY, USA, 2010.

[21] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst, 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*, Stroudsburg, PA, USA, 2007, pp. 177–180.

[22] R. Sennrich, Perplexity minimization for translation model domain adaptation in statistical machine translation, *In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, 2012, pp. 539–549.

[23] R. Zens, F.J. Och and H. Ney, Phrase-based statistical machine translation, In: KI., 2002, pp. 18–32.

[24] S.F. Chen and J. Goodman, An empirical study of smoothing techniques for language modeling, *In: Proceedings of the 34th annual meeting on Association for Computational Linguistics ACL '96*, Stroudsburg, PA, USA, Association for Computational Linguistics, 1996, pp. 310–318.