Neural Probabilistic Language Model for System Combination

Tsuyoshi Okita[1]
(1) Dublin City University, Glasnevin, Dublin 9
tokita@computing.dcu.ie

ABSTRACT
This paper gives the system description of the neural probabilistic language modeling (NPLM) team of Dublin City University for our participation in the system combination task in the Second Workshop on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid MT (ML4HMT-12). We used the information obtained by NPLM as meta information to the system combination module. For the Spanish-English data, our paraphrasing approach achieved 25.81 BLEU points, which lost 0.19 BLEU points absolute compared to the standard confusion network-based system combination. We note that our current usage of NPLM is very limited due to the difficulty in combining NPLM and system combination.

KEYWORDS: statistical machine translation, neural probabilistic language model, system combination.

1 Introduction

This paper describes a new extension to our system combination module developed in Dublin City University (Du and Way, 2010a,b; Okita and van Genabith, 2012). We have added a neural probabilistic language model (NPLM) (Bengio et al., 2000, 2005) to our system combination module and tested it in the system combination task at the ML4HMT-2012 workshop.

A neural probabilistic language model (NPLM) (Bengio et al., 2000, 2005) and the distributed representations (Hinton et al., 1986) provide an idea to achieve the better perplexity than n-gram language model (Stolcke, 2002) and their smoothed language models (Kneser and Ney, 1995; Chen and Goodman, 1998; Teh, 2006). Recently, the latter one, i.e. smoothed language model, has had a lot of developments in the line of nonparametric Bayesian methods such as hierarchical Pitman-Yor language model (HPYLM) (Teh, 2006) and Sequence Memoizer (SM) (Wood et al., 2009; Gasthaus et al., 2010), including an application to SMT (Okita and Way, 2010a,b, 2011). A NPLM considers the representation of data in order to make the probability distribution of word sequences more compact where we focus on the similar semantical and syntactical roles of words. For example, when we have two sentences *"The cat is walking in the bedroom"* and *"A dog was running in a room"*, these sentences can be more compactly stored than the n-gram language model if we focus on the similarity between (the, a), (bedroom, room), (is, was), and (running, walking). Thus, a NPLM provides the semantical and syntactical roles of words as a language model. A NPLM of Bengio et al. (2000) implemented this using the multi-layer neural network and yielded 20% to 35% better perplexity than the language model with the modified Kneser-Ney methods (Chen and Goodman, 1998).

There are several successful applications of NPLM (Schwenk, 2007; Collobert and Weston, 2008; Schwenk, 2010; Collobert, 2011; Collobert et al., 2011; Deschacht et al., 2012; Schwenk et al., 2012). First, one category of applications include POS tagging, NER tagging, and parsing (Collobert et al., 2011; Bordes et al., 2011). This category uses the features provided by a NPLM in the limited window size.[1] It is often the case that there is no such long range effects that the decision cannot be made beyond the limited windows which requires to look carefully the elements in a long distance. Second, the other category of applications include Semantic Role Labeling (SRL) task (Collobert et al., 2011; Deschacht et al., 2012). This category uses the features within a sentence. A typical element is the predicate in a SRL task which requires the information which sometimes in a long distance but within a sentence. Both of these approaches do not require to obtain the best tag sequence, but these tags are independent. Third, the final category includes MERT process (Schwenk, 2010) and possibly many others where most of them remain undeveloped. The objective of this learning in this category is not to search the best tag for a word but the best sequence for a sentence. Hence, we need to apply the sequential learning approach.[2] Although most of the applications described in (Collobert and Weston, 2008; Collobert, 2011; Collobert et al., 2011; Deschacht et al., 2012) are monolingual tasks, the application of this approach to a bilingual task introduces really astonishing aspects, which we can call "creative words" (Veale, 2012), automatically into the traditional resource constrained SMT components. For example, the training corpus of word aligner is often strictly restricted to the

[1] It is possible to implement a parser in the way of the second category. However, we adopt the categorization which was implemented by (Collobert et al., 2011).

[2] The first and second approaches do not often appear in the context of SMT, while the third category includes most of the decoding algorithm appeared in SMT including MAP decoding, MBR decoding, and (monotonic) consensus decoding. The latter two decoding appears in system combination.

given parallel corpus. However, a NPLM allows this training with huge monolingual corpus. Although most of this line has not been even tested mostly due to the problem of computational complexity of training NPLM, Schwenk et al. (2012) applied this to MERT process which reranks the n-best lists using NPLM. This paper aims at different task, a task of system combination (Bangalore et al., 2001; Matusov et al., 2006; Tromble et al., 2008; Du et al., 2009; DeNero et al., 2009; Okita and van Genabith, 2012). This category of tasks employs the sequential method such as Maximum A Posteriori (MAP) inference (Viterbi decoding) (Koller and Friedman, 2009; Sontag, 2010; Murphy, 2012) on Conditional Random Fields (CRFs) / Markov Random Fields (MRFs). [3]

The remainder of this paper is organized as follows. Section 2 describes our algorithms. In Section 3, our experimental results are presented. We conclude in Section 4.

2 Our Algorithms

The aim of NPLM is to capture the semantically and syntactically similar words in a way that a latent word depends on the context. There would be many ways to use this language model. However, one difficulty resides in how such information can be incorporated to the module of system combination. Due to this difficulty, we present here a method which is rather restricted despite the power of NPLM.

This paper presents two methods based on the intuitive observation that we will get the variety of words if we condition on the fixed context, which would form paraphrases in theory. Then, we present the second method that we examine the dependency structure of candidates sentence replaced with alternative expressions (or paraphrases). Our algorithms consist of three steps shown in Algorithm 1. The details of Step 1 and 2 will be explained in Section 2.1 and Step 3 will be explained in Section 2.2.

Algorithm 1 Our Algorithm

Given: For given testset g, prepare N translation outputs $\{s_1, \ldots, s_N\}$ from several systems.
Step 1: Train NPLM with monolingual corpus. Note that this monolingual corpus would be better in the same domain as the testset g.
Step 2: Modify the translation outputs $\{s_1, \ldots, s_N\}$ replaced with alternative expressions (or paraphrases).
Step 3: Augment the sentences of translation outputs prepared in Step 2.
Step 4: Run the system combination module.

2.1 Paraphrasing using NPLM

2.1.1 Plain Paraphrasing

We introduce our algorithm via a word sense disambiguation (WSD) task which selects the right disambiguated sense for the word in question. This task is necessary due to the fact that a text is natively ambiguous accommodating with several different meanings. The task of WSD (Deschacht et al., 2012) can be written as in (1):

$$P(\text{synset}_i|\text{features}_i, \theta) = \frac{1}{Z(\text{features})} \prod_m g(\text{synset}_i, k)^{f(\text{feature}_i^k)} \quad (1)$$

[3] Note that the (monotonic) consensus decoding in system combination is the subset of this.

where k ranges over all possible features, $f(\text{feature}_i^k)$ is an indicator function whose value is 1 if the feature exists, and 0 otherwise, $g(\text{synset}_i, k)$ is a parameter for a given synset and feature, θ is a collection of all these parameters in $g(\text{synset}_i, k)$, and Z is a normalization constant. Note that we use the term "synset" as an analogy of the WordNet (Miller, 1995): this is equivalent to "sense" or "meaning". Note also that NPLM will be included as one of the features in this equation. If features include sufficient statistics, a task of WSD will succeed. Otherwise, it will fail.

Now we assume that the above WSD component was trained. We would like to consider the paraphrasing in connection with this. We consider a sentence with some words replaced by the alternative surface form. In this context, we are interested in the words which share the same synset (or meaning) but the realized surface form is different. Let us denote $P(\text{surface}_i|\text{synset}_j, \text{features}_k, \theta)$ by the probability of such words. Then, we suppose that we compare $\text{P}(\text{surface}_i{=}x_i\,|\,\text{synset}_j,\text{features}_k,\theta)$ and $\text{P}(\text{surface}_i{=}x_i'\,|\,\text{synset}_j,\text{features}_k,\theta)$ under the condition that synset_j, features_k, and θ are the same, and that the relationships below hold as in (2):

$$P(\text{surface}_i = x_i|\text{synset}_j, \text{features}_k, \theta) > P(\text{surface}_i = x_i'|\text{synset}_j, \text{features}_k, \theta) \tag{2}$$

Then, the alternative surfaces form x_i in higher probability will be chosen instead of the other one x_i' among paraphrases $\{x_i, x_i'\}$ of this word.

On the one hand, the paraphrases obtained in this way have attractive aspects that can be called "a creative word" (Veale, 2012). This is since the traditional resource that can be used when building a translation model by SMT are constrained on paralell corpus. However, NPLM can be trained on huge monolingual corpus. On the other hand, unfortunately in practice, the notorious training time of NPLM only allows us to use fairly small monolingual corpus although many papers made an effort to reduce it (Mnih and Teh, 2012). Due to this, we cannot ignore the fact that NPLM trained not on a huge corpus may be affected by noise. Conversely, we have no guarantee that such noise will be reduced if we train NPLM on a huge corpus. It is quite likely that NPLM has a lot of noise for small corpora. Hence, this paper also needs to provide the way to overcome difficulties of noisy data. In order to avoid this difficulty, we limit the paraphrase only when it includes itself in high probability.

2.1.2 Paraphrasing with Modified Dependency Score

Although we modified a suggested paraphrase without any intervention in the above algorithm, it is also possible to examine whether such suggestion should be adopted or not. If we add paraphrases and the resulted sentence has a higher score in terms of the modified dependency score (Owczarzak et al., 2007) (See Figure 1), this means that the addition of paraphrases is a good choice. If the resulted score decreases, we do not need to add them. One difficulty in this approach is that we do not have a reference which allows us to score it in the usual manner. For this reason, we adopt the *naive way* to deploy the above and we deploy this with *pseudo references*. First, if we add paraphrases and the resulted sentence does not have a very bad score, we add these paraphrases since these paraphrase are not very bad (*naive* way). Second, we do scoring between the sentence in question with *all the other candidates* (*pseudo references*) and calculate an average of them. Thus, our second algorithm is to select a paraphrase which may not achieve a very bad score in terms of the modified dependency score using NPLM.

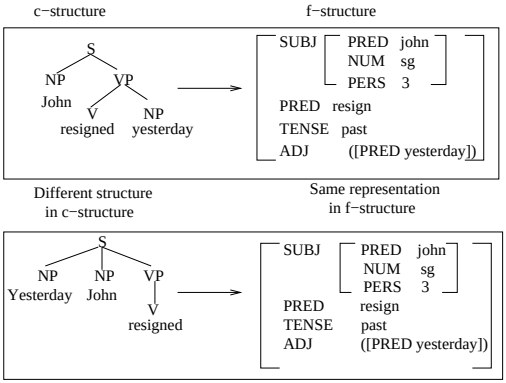

Figure 1: By the modified dependency score (Owczarzak et al., 2007), the score of these two sentences, "John resigned yesterday" and "Yesterday John resigned", are the same. Figure shows c-structure and f-structure of two sentences using Lexical Functional Grammar (LFG) (Bresnan, 2001).

Modified Dependency Score We mention here the reason why we use the modified dependency score. First, unlike BLEU or NIST, the modified dependency score can capture the syntactical construction and grammaticality of sentences. Second, our current dataset seems to be interesting since it includes Lucy LT RBMT outputs. (The last year's ML4HMT-11 (Okita and van Genabith, 2011) also included the translation output of Lucy LT RBMT outputs.) If we choose the entire Lucy's output as a backbone and run a system combination, the resulted score is the highest among various system combination strategies we tried (See "s2 backbone" in Table 2). These two facts suggest us the following strategy: if we have prior knowledge that the Lucy backbone will obtain a high score, it would be interesting to start from the Lucy backbone and pursue whether we can improve the overall score further by adding paraphrases. As is evident from the fact that the Lucy backbone is good, our interest will not be BLEU or NIST, but MODIFIED DEPENDENCY SCORE. This may lead to a higher BLEU score than the system combination results with Lucy backbone. Note that in order to make it a universal algorithm, we need to remove Lucy backbone from this algorithm. Hence, only the modified dependency score remains, which forms the algorithm already mentioned.

system	translation output	precision	recall	F-score
s1	these do usually in a week .	0.080	0.154	0.105
s2	these are normally made in a week .	0.200	0.263	0.227
s3	they are normally in one week .	0.080	0.154	0.105
s4	they are normally on a week .	0.120	0.231	0.158
ref	the funding is usually offered over a one-week period .			

Table 1: An example of modified dependency score for a set of translation outputs.

2.2 System Combination

As is mentioned at the beginning of this section, the interface between the NPLM and system combination has some difficulties. This contrasts with the task of n-best reranking

(Schwenk et al., 2012). In the case of n-best reranking, the probability provided by NPLM can be used immediately in the re-evaluation of the n-best lists.

Difficulties of Interface In our case, due to the reason below despite the advantage of word varieties it is difficult to incorporate this into the translation outputs in a straight forward way. The two decoding strategies used by a confusion network-based system combination, i.e. MBR decoding and (monotonic) consensus decoding, have difficulties in each step.

First, in MBR decoding in the first step, the inputs, i.e. each translation outputs, are quantified by the loss function with its score in the sentence level. This mechanism does not allow us to add fragments freely in the word level. Therefore, it requires us to increase the number of sentences with only a slight replacement in the sentence level. This paper takes this strategy for the first step to circumvent this difficulty.

Second, in (monotonic) consensus decoding in the second step, the word posterior probabilities in the confusion network do not reflect the probability quantified globally, but is rather locally in accordance with other probability of words in the same position. Similarly, one way would be to augment in the sentence level.

Inputs to System Combination Module We check the possibilities whether the word can have alternative expression and whether the probability of such expression is bigger than that of the original word or not. If this holds, we replace such words with alternative expressions. This will make a new sentence.

3 Experimental Results

ML4HMT-2012 provides four translation outputs (*s1* to *s4*) which are MT outputs by two RBMT systems, APERTIUM and LUCY, PB-SMT (MOSES) and HPB-SMT (MOSES), respectively. The tuning data consists of 20,000 sentence pairs, while the test data consists of 3,003 sentence pairs.

Our experimental setting is as follows. We use our system combination module (Du and Way, 2010a,b; Okita and van Genabith, 2012), which has its own language modeling tool, MERT process, and MBR decoding. We use the BLEU metric as loss function in MBR decoding. We use TERp[4] as alignment metrics in monolingual word alignment. We trained NPLM using 500,000 sentence pairs from English side of EN-ES corpus of EUROPARL[5].

(a)	*the Government wants to limit the torture of the " witches " , as it published in a brochure*
(b)	the Government wants to limit the torture of the " witches " , as it published in the proceedings
(a)	*the women that he " return " witches are sent to an area isolated , so that they do not hamper the rest of the people .*
(b)	the women that he " return " witches are sent to an area eligible , so that they do not affect the rest of the country .

Table 2: Table includes two examples of plain paraphrase.

The results show that the first algorithm of NPLM-based paraphrased augmentation, that is

[4] http://www.cs.umd.edu/~snover/terp

[5] http://www.statmt.org/europarl

NPLM plain, achieved 25.61 BLEU points, which lost 0.39 BLEU points absolute over the standard system combination. The second algorithm, NPLM dep, achieved slightly better results of 25.81 BLEU points, which lost 0.19 BLEU points absolute over the standard system combination. Note that the baseline achieved 26.00 BLEU points, the best single system in terms of BLEU was s4 which achieved 25.31 BLEU points, and the best single system in terms of METEOR was s2 which achieved 0.5853.

	NIST	BLEU	METEOR	WER	PER
s1	6.4996	0.2248	0.5458641	64.2452	49.9806
s2	6.9281	0.2500	0.5853446	62.9194	48.0065
s3	7.4022	0.2446	0.5544660	58.0752	44.0221
s4	7.2100	0.2531	0.5596933	59.3930	44.5230
NPLM plain	7.6041	0.2561	0.5593901	56.4620	41.8076
NPLM dep	7.6213	0.2581	0.5601121	56.1334	41.7820
BLEU-MBR	7.6846	0.2600	0.5643944	56.2368	41.5399
min ave TER-MBR	7.6231	0.2638	0.5652795	56.3967	41.6092
DA	7.7146	0.2633	0.5647685	55.8612	41.7264
QE	7.6846	0.2620	0.5642806	56.0051	41.5226
s2 backbone	7.6371	0.2648	0.5606801	56.0077	42.0075
modDep precision	7.6670	0.2636	0.5659757	56.4393	41.4986
modDep recall	7.6695	0.2642	0.5664320	56.5059	41.5013
modDep Fscore	7.6695	0.2642	0.5664320	56.5059	41.5013

	modDep precision	modDep recall	modDep Fscore
average s1	0.244 (586)	0.208	0.225
average s2	0.250 (710)	0.188	0.217
average s3	0.189 (704)	0.145	0.165
average s4	0.195 (674)	0.167	0.180

Table 3: This table shows single best performance, the performance of two algorithms in this paper (NPLM plain and dep), MBR-decoding with BLEU loss function and TER loss function, the performance of domain adaptation (Okita et al., 2012b) and quality estimation (Okita et al., 2012a), the performance of Lucy backbone, and the performance of the selection of sentences by modified dependency score (precision, recall, and F-score each). The four lines at the bottom marked with average s1 to s4 indicates the average performance of s1 in terms of precision, recall, and F-score (from the 2nd to 4th columns) when we make the backbone by choosing the maximum score in terms of the modified dependency score. For example, the first line of "modDep precision" shows when we chose a backbone by the maximum modified dependency score in terms of precision. 586 sentences were selected from s1, 710 sentences were from s2, and so forth. The average BLEU score of these 586 sentences was 24.4.

Conclusion and Perspectives

This paper deployed meta information obtained by NPLM into a system combination module. NPLM captures the semantically and syntactically similar words in a way that a latent word depends on the context. First, we interpret the information obtained by NPLM as paraphrases with regard to the translation outputs. Then, we incorporate the augmented sentences as inputs to the system combination module. Unfortunately, this strategy lost 0.39 BLEU points

absolute compared to the standard confusion network-based system combination. A revised strategy to assess the quality of paraphrases achieved 25.81 BLEU points, which lost 0.19 BLEU points absolute.

There are many further avenues. First, as already mentioned, this paper only scratched the surface of NPLM. One problem was the interface between NPLM and system combination. Our motivation behind using NPLM was the possibility that NPLM would supply the semantically and syntactically rich synonyms and similar words to the rather restricted translation outputs, as well as the traditional functions as LM, which are to be supplied to the system combination module. For this reason, we believe that paraphrases generated using NPLM will not be a bad direction. However, there would be other approach as well. Collobert and Weston (2008) and Bordes et al. (2011) integrate NPLM in their software. When we integrate our approach, one way would be to implement it without employing the knowledge of paraphrases. It would be interesting to compare this and our approaches in this paper. Alternatively, prior knowledge about the speriority of Lucy output can be embedded into system combination by prior (Okita et al., 2010b,a; Okita, 2012).

Second, we show some positive results about the modified dependency score (Owczarzak et al., 2007). We used this as sentence-based criteria to select a backbone in three ways: maximum precision, recall, and F-score. Results are shown in Table 3. Indeed, these criteria worked quite well. Unfortunately, these scores were still lower than that of Lucy's backbone. The lower parts of this table show statistics when we select a backbone by the modified dependency score. Interestingly, the modified dependency score of s2 (Lucy) was the best in precision score, but was not the best in recall or in F-score. This shows that the selection of backbone by the modified dependency score did not work as much as that of the (fixed) Lucy backbone. We need to search another explanation why the Lucy backbone obtained the highest score.

Third, this paper did not mention much about noise. Understanding the mechanism of noise on NPLM may be related to the learning mechanism of NPLM, if we draw an analogy from the case when we examined the noise of word alignment (Okita, 2009; Okita et al., 2010b). This may also be related to the smoothing mechanism (Okita and Way, 2010a,b, 2011).

Acknowledgments

We thank Maite Melero for proof-reading. This research is partly supported by the 7th Framework Programme and the ICT Policy Support Programme of the European Commission through the T4ME (Grant agreement No. 249119) project as well as by Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

References

Bangalore, S., Bordel, G., and Riccardi, G. (2001). Computing consensus translation from multiple machine translation systems. *In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 350–354.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *In Proceedings of Neural Information Systems*.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2005). Neural probabilistic language models. *Innovations in Machine Learning: Theory and Applications Edited by D. Holmes and L. C. Jain*.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2011). Towards open-text semantic parsing via multi-task learning of structured embeddings. *CoRR*, abs/1107.3663.

Bresnan, J. (2001). Lexical functional syntax. *Blackwell*.

Chen, S. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. *Technical report TR-10-98 Harvard University*.

Collobert, R. (2011). Deep learning for efficient discriminative parsing. *In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *In International Conference on Machine Learning (ICML 2008)*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

DeNero, J., Chiang, D., and Knight, K. (2009). Fast consensus decoding over translation forests. *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 567–575.

Deschacht, K., Belder, J. D., and Moens, M.-F. (2012). The latent words language model. *Computer Speech and Language*, 26:384–409.

Du, J., He, Y., Penkale, S., and Way, A. (2009). MaTrEx: the DCU MT System for WMT 2009. *In Proceedings of the Third EACL Workshop on Statistical Machine Translation*, pages 95–99.

Du, J. and Way, A. (2010a). An incremental three-pass system combination framework by combining multiple hypothesis alignment methods. *International Journal of Asian Language Processing*, 20(1):1–15.

Du, J. and Way, A. (2010b). Using terp to augment the system combination for smt. *In Proceedings of the Ninth Conference of the Association for Machine Translation (AMTA2010)*.

Gasthaus, J., Wood, F., and Teh, Y. W. (2010). Lossless compression based on the sequence memoizer. *DCC 2010*.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. (1986). Distributed representations. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition(Edited by D.E. Rumelhart and J.L. McClelland) MIT Press*, 1.

Kneser, R. and Ney, H. (1995). Improved backing-off for n-gram language modeling. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184.

Koller, D. and Friedman, N. (2009). Probabilistic graphical models: Principles and techniques. *MIT Press*.

Matusov, E., Ueffing, N., and Ney, H. (2006). Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 33–40.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mnih, A. and Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference on Machine Learning*.

Murphy, K. P. (2012). Machine learning: A probabilistic perspective. *The MIT Press*.

Okita, T. (2009). Data cleaning for word alignment. *In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009) Student Research Workshop*, pages 72–80.

Okita, T. (2012). Annotated corpora for word alignment between japanese and english and its evaluation w ith map-based word aligner. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evalu ation (LREC-2012)*, pages 3241–3248, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1655.

Okita, T., Graham, Y., and Way, A. (2010a). Gap between theory and practice: Noise sensitive word alignment in machine translation. *In Proceedings of the Workshop on Applications of Pattern Analysis (WAPA2010). Cumberland Lodge, England.*

Okita, T., Guerra, A. M., Graham, Y., and Way, A. (2010b). Multi-Word Expression sensitive word alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010, collocated with COLING2010), Beijing, China.*, pages 1–8.

Okita, T., Rubino, R., and van Genabith, J. (2012a). Sentence-level quality estimation for mt system combination. *In Proceedings of ML4HMT Workshop (collocated with COLING 2012)*.

Okita, T., Toral, A., and van Genabith, J. (2012b). Topic modeling-based domain adaptation for system combination. *In Proceedings of ML4HMT Workshop (collocated with COLING 2012)*.

Okita, T. and van Genabith, J. (2011). DCU Confusion Network-based System Combination for ML4HMT. *Shared Task on Applying Machine Learning techniques to optimising the division of labour in Hybrid MT (ML4HMT-2011, collocated with LIHMT-2011)*.

Okita, T. and van Genabith, J. (2012). Minimum bayes risk decoding with enlarged hypothesis space in system combination. *In Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2012). LNCS 7182 Part II. A. Gelbukh (Ed.)*, pages 40–51.

Okita, T. and Way, A. (2010a). Hierarchical pitman-yor language model in machine translation. *In Proceedings of the International Conference on Asian Language Processing (IALP 2010)*.

Okita, T. and Way, A. (2010b). Pitman-Yor process-based language model for Machine Translation. *International Journal on Asian Language Processing*, 21(2):57–70.

Okita, T. and Way, A. (2011). Given bilingual terminology in statistical machine translation: Mwe-sensitve word alignment and hierarchical pitman-yor process-based translation model smoothing. *In Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference (FLAIRS-24)*, pages 269–274.

Owczarzak, K., van Genabith, J., and Way, A. (2007). Evaluating machine translation with LFG dependencies. *Machine Translation*, 21(2):95–119.

Schwenk, H. (2007). Continuous space language models. *Computer Speech and Language*, 21:492–518.

Schwenk, H. (2010). Continuous space language models for statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 83:137–146.

Schwenk, H., Rousseau, A., and Attik, M. (2012). Large, pruned or continuous space language models on a gpu for statistical machine translation. *In Proceeding of the NAACL workshop on the Future of Language Modeling*.

Sontag, D. (2010). Approximate inference in graphical models using LP relaxations. *Massachusetts Institute of Technology (Ph.D. thesis)*.

Stolcke, A. (2002). SRILM – An extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Teh, Y. W. (2006). A hierarchical bayesian language model based on pitman-yor processes. *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL-06), Prague, Czech Republic*, pages 985–992.

Tromble, R., Kumar, S., Och, F., and Macherey, W. (2008). Lattice minimum bayes-risk decoding for statistical machine translation. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629.

Veale, T. (2012). Exploding the creativity myth: The computational foundations of linguistic creativity. *London: Bloomsbury Academic*.

Wood, F., Archambeau, C., Gasthaus, J., James, L., and Teh, Y. W. (2009). A stochastic memoizer for sequence data. *In Proceedings of the 26th International Conference on Machine Learning*, pages 1129–1136.