Improving Low-Resource Statistical Machine Translation with a Novel Semantic Word Clustering Algorithm

Jeff MA
50 Moulton St, Cambridge, MA, USA
jma@bbn.com

Spyros Matsoukas
50 Moulton St, Cambridge, MA, USA
smatsouk@bbn.com

Richard Schwartz
50 Moulton St, Cambridge, MA, USA
Schwartz@bbn.com

Abstract

In this paper we present a non-language-specific strategy that uses large amounts of monolingual data to improve statistical machine translation (SMT) when only a small parallel training corpus is available. This strategy uses word classes derived from monolingual text data to improve the word alignment quality, which generally deteriorates significantly because of insufficient training. We present a novel semantic word clustering algorithm to generate the word classes motivated by the word similarity metric presented in (Lin, 1998). Our clustering results showed this novel word clustering outperforms a state-of-the-art hierarchical clustering. We then designed a new procedure for using the derived word classes to improve word alignment quality. Our experiments showed that the use of the word classes can recover over 90% of the loss resulting from the alignment quality that is lost due to the limited parallel training.

1 Introduction

Relatively little machine translation (MT) research has been reported on languages that lack resources, such as monolingual text, parallel text, translation dictionaries, syntactic and semantic parsing tools, which in the literature are often referred to as "low-density" or "low-resource". Nowadays it becomes much easier to obtain monolingual text data through the internet, but it still remains difficult to obtain parallel text. In this paper we deal with a low-resource situation where only a limited amount of bilingual text is available but large amounts of monolingual text are available for both the source and target languages. One more important reason to begin with such a situation was that we would like to explore approaches that can utilize the availability of large amounts of monolingual data for improving SMT.

1.1 Related work

Maxwell and Hughes (2006) proposed two strategies for low-resource languages: (I) transfer relevant linguistic information from existing tools and resources from resource-rich languages to a lower-resource language, and (II) develop methods that require less data. Strategy (I) is often used between two closely related languages. For example, the bootstrapping of an Urdu-English MT with a Hindi-English system in (Sinha, 2009) and generating rules for the Yiddish-English MT from the German-English and Hebrew-English MT in (Genzel et al., 2009). There are more efforts reported on exploring strategy (II), such as the work in (Al-Onaizan et al., 2000, Nießen and Ney, 2004, Roy and Popwich, 2010, Wang et al., 2006, Homola and Kubon, 2005, Xiang et al., 2010). All those methods are language-specific, because they either need to find the relevant resource-rich language or use language-specific features. Baker et al. (2010) showed more general ways to incorporate syntactic and semantic knowledge into the Urdu-English MT systems, but the knowledge was obtained through parsing both the source and target languages. Parsing tools are often unavailable in low-resource situations. Thus, all those methods cannot be easily applied to different languages. We aimed at exploring approaches that can be easily applied to any languages.

1.2 Our strategy and contribution

The performance of SMT normally degrades greatly when training data becomes insufficient. The degradation mainly results from two factors, less reliable translation rules and poorer rule coverage. In this paper we focus only on the first

Approved for Public Release, Distribution Unlimited. – June, 2011

factor. Word alignment quality is one of the biggest factors that affect the extraction of reliable translation rules. Statistical word aligners, such as GIZA++ (Och and Ney, 2003), are generally weak at aligning infrequent words properly because of the insufficient statistical evidence. With improper word alignments unreliable translation rules are extracted. In this paper we focus on improving word alignment quality. The statistical evidence for word classes, in general are more adequate, so alignments estimated for word classes would be relatively more reliable. Our strategy was to use word class alignments to improve word alignments. Some work has been reported that improved MT with word classes, such as generating syntactic and semantic features for the SMT decoding in (Baker et al. 2010), broadening the coverage of word alignments with class alignments in (Ker and Zhang, 1997) and improving word alignments with cross-lingual word similarity in (Wang and Chou, 2004). All those methods are more or less language-specific, and could not easily be applied to other languages. Our major contribution here is the development of a novel semantic word clustering algorithm that can be easily applied to any language. Another contribution is the design of a new procedure for using word classes to improve word alignment quality. Our results showed that the new algorithm outperforms a start-of-the-art hierarchical clustering approach when used to improve word alignment quality.

We organize this paper as follows: Section 2 describes the experimental setup; Section 3 presents the novel semantic word clustering algorithm; and Section 4 addresses the strategy that uses word classes to improve word alignment quality.

2 Experimental Setup

As mentioned before, we began with the low-resource situation where only a limited amount of bilingual text is available but large amounts of monolingual text are available for both source and target languages. Since we aimed at developing strategies that are not language-specific, we had no constraints on the selection of the two languages. We thus decided to simulate the situation with two languages that *do have* a great amount of parallel training data available. In this way, we can compare and measure the performance of our strategies against the situation where sufficient parallel training data is available. Because of the availability of a large amount Chinese-English bilingual text, we selected Chinese-to-English MT to experiment with.

2.1 Training data

The Linguistic Data Consortium (LDC) has released a number of Chinese-English bi-text collections[1] that include millions of words for the purpose of MT evaluation. With these collections we had set up a 200 million (200M) word [2] parallel training corpus. To simulate the low-resource condition, we downsized the training data to 125 thousand (125K) words. These 125K words were randomly extracted from two newswire collections – LDC2005T10 and LDC2006G05. We assumed that it would be reasonable to have a 40K-50K word translation dictionary for building a MT system for any low-resource language, so we added a 40K Chinese-English word translation dictionary to the 125K training corpus. We acquired this 40K dictionary through the ADSO project[3]. This dictionary was also included in the "200M" training corpus. Hence, the limited training includes 165K words. We will use "165K" to denote this limited training corpus.

Since we were not limiting the monolingual resources, we used a large amount of data to train the English language model (LM). The LM training data consists of 2.2 billion (2.2B) words from the 4th edition of the LDC English Gigaword monolingual data release, 2.5 billion (2.5B) words from Google news and 1.6 billion words from web news that we downloaded from various websites, such as BBC, XinHua News, NewsArchives and The Arab News. The 165K English words from the "165K" parallel training were also included in the LM training. The total number of words was 6.4 billion (6.4B).

2.2 Test data

For system tuning in the low-resource situation, we assumed that it is affordable to hire one bilingual expert to translate 1000 (1K) sentences. To set up such a tuning set, we randomly extracted 1,000 sentences from the newswire portion of the NIST MT04, MT05, MT06, and MT08 evalua-

[1] http://projects.ldc.upenn.edu/gale/data/catalog.htm
[2] We counted the size of a training corpse as the number of target words.
[3] The ADSO dictionary is not really a word translation dictionary, because it also includes phrase translations. The 40K dictionary we used actually consisted of 25K entries and 40K English words. The latest release (v5.077) of the ADSO dictionary consists of 185K entries, which is free to public (http://www.adsotrans.com/downloads).

tion sets[4]. Some of the 1K sentences have multiple (usually 4) reference translations. For these sentences we kept only one reference translation (chosen randomly). For measuring the MT performance we randomly selected 3,000 (3K) sentences from the same newswire portion without overlaps with the 1K tuning set. Many of the 3K sentences have multiple reference translations, which we thought would be fine to be kept for better score measuring. We used the IBM BLEU (Papineni et al., 2002) metric to measure the MT performance. All the scores reported in this paper were measured on the 3K test set.

2.3 The baseline performance

We built our SMT systems based on the model described in (Shen et al., 2008). We used GIZA++ to train word alignments. The decoding parameters were tuned using the minimum error rate method (Och, 2003) to maximize the BLEU score. The BLEU scores for the MT models trained with the 200M and the 165K training corpuses are shown in Table 1. As can be seen, there is a big performance loss (17.3 = 35.4-18.1) when the amount of training is reduced from 200M to 165K (column "200M" vs. column "165K"). The "165K" system serves as the baseline for the work reported in this paper.

Train data	200M	165K	165K-best
BLEU	35.37	18.06	20.36

Table 1. MT performance (BLEU scores) of systems trained with different amounts of data

What was the maximum gain that we could obtain from the effort of improving the word alignment quality for the "165K" training data? To answer this question we would need the perfect alignments, which are impossible to acquire in practice. When training data is sufficient, GIZA++ is able to produce good quality of word alignments. Since the "165K" training data was a subset of the 200M parallel training corpus, we extracted word alignments for the "165K" training corpus from the alignments trained with the 200M corpus. It is fairly reasonable to assume the extracted alignments were the "best" alignments that we could get from using any of the statistical word aligners. The BLEU score with the "best" alignments is also shown in Table 1., denoted as "165K-best". Compared to the baseline, the "best" alignments improved the MT by 2.3 BLEU points (=20.36-18.06). So, the maximum gain (or the upper bound) we could acquire from improving the alignment quality was 2.3 points.

On the other hand, the 2.3 point loss, caused by the deterioration of word alignment quality, was only a small fraction of the 17 point loss when the amount of training was reduced from 200M to 165K, so the majority of the loss was from the other factor – the poorer rule coverage, which will be our focus in the future.

3 A novel word clustering algorithm

For MT, source and target words that have the same meaning should be aligned together. So it is better to cluster words semantically if word class alignments are used to improve word alignments. Ker and Zhang (1997) used man-made thesauri in their work to help aligning words. Thesauri are often unavailable in low-resource situations. Hence, to achieve our goal we needed a semantic clustering algorithm, which clusters synonyms together.

A number of semantic clustering algorithms have been reported, such as those in (Bellegarda et al., 1996, Geffet and Dagan, 2004, Ichioka and Fukmoto, 2008, Lee, 1999, Lin, 1998, Muller et al., 2006, Sinha and Mihalcea, 2007, Weeds and Weir, 2005). But we found the word similarity measure reported in (Lin, 1998) was more attractive, because we could easily generalize it to develop a word clustering algorithm that can be used on any language. In the paper the mutual information between two words w_1 and w_2 is defined as,

$$I(w_1, r, w_2) = \log \frac{Cnt(w_1, r, w_2) \cdot Cnt(*, r, *)}{Cnt(w_1, r, *) \cdot Cnt(*, r, w_2)}$$

where "r" represents the grammatical relationship of w_1 and w_2, such as "is subject of", "is object of" and $Cnt(\cdot)$ denotes the count.

The similarity between two words w_1 and w_2 is then computed based on the mutual information,

$$sim(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \cap T(w_2)} [I(w_1, r, w) + I(w_2, r, w)]}{\sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w)}$$

[4] http://www.ldc.upenn.edu/Catalog/project_index.jsp

where $T(w_1)$ represents all pairs of *(r,w)* that makes $I(w_1, r, w)$ positive and $T(w_2)$ **is** all pairs of *(r,w)* that makes $I(w_2, r, w)$ positive.

Lin (1998) defined tens of typical grammatical relationships and used them in the similarity computation. The grammatical relationships were obtained by parsing all the training sentences. We could not do the same thing here because there are no parsing tools available in low-resource situations. To avoid using the grammatical relationships, we generalized the similarity computation by replacing "*r*" with n-grams. Since it is infeasible to consider all orders of n-grams that have ever occurred between two words w_1 and w_2, we only used {0,1,2,3}-grams and ignored all the n-grams that are higher than the 3rd order. The "0-gram" here means there are no words between w_1 and w_2. With this generalization, $Cnt(w_1, r, w_2)$ is thus changed to be the total counts of {2,3,4,5}-grams that start with w_1 and end with w_2, $T(w_1)$ to be all the {1,2,3,4}-grams that have occurred immediately after word "w_1", and all others changed accordingly.

With the similarity between words computed, we did a bottom-up clustering to cluster the words as follows,

1) Initialize each individual word as a class
2) Compute the similarity between any two classes
3) Rank the class-to-class similarities from the highest to the lowest, and merge the top-n class pairs.
4) Stop if the desired number of clusters is obtained. Otherwise, go to Step 2.

In Step 2), we computed the similarity between two classes C_1 and C_2 according to

$$sim(C_1, C_2) = \frac{1}{N_1 * N_2} \sum_{w_1 \in C_1} \sum_{w_2 \in C_2} sim(w_1, w_2) + \frac{\lambda}{N_1 + N_2}$$

where N_1 and N_2 denote the numbers of words in the classes, C_1 and C_2, respectively. We added the term $\frac{\lambda}{N_1 + N_2}$ to the class similarity computation, tending to have a higher priority for smaller classes to be merged. In our experiments we set λ to 0.2. In Step 3) it would be best to merge only the one best pair of classes at each iteration, but we found this was too slow, so we merge top-2,000 class pairs at each iteration. If a class had been merged, we skipped all the lower-ranked class pairs in which it occurred again. We denote this word-similarity-based clustering algorithm as "WSB clustering".

3.1 Clustering English words

The English monolingual text corpus that we used to train word classes consisted of 5B words. This corpus is a subset of the 6.4B LM training corpus, described in Section 2.1. It consists of the 200M English words from the 200M Chinese-English parallel corpus, the 2.2B English Gigaword words, and the 2.5B Google news words. After some cleanup – mainly removing words that occur only once or twice, there are about 1.1M unique words in the corpus. We then ran the WSB clustering to produce 5K, 20K and 50K classes.

Hierarchical clustering	WSB clustering
obama clack granof vaduva outzen lackey robold mordashow	obama hillary biden bush lieberman gore mccain clinton kerry palin
michael steven philip hezekiah dunstan alayne lydia	michael john robert bob chris mike peter david
massachusetts queenstown farmington sakha eurasia uil johor	massachusetts montana vermont delaware nevada wyoming idaho dakota maine
lawyer reviewer investigator privateer baldish sideman	lawyer prosecutor attorney counsel
christianity islam solvang cyberzone pointillism ngoya	christianity catholicism taoism orthodoxy hinduism judaism confucianism daoism buddhism kaaba
announced revealed mentioned announces ruled editorialized presupposed envisaged reapproved	announced announce announces unveiling unveils announcing unveiled unveil
transfer disapprove retransmissions retransmission indi regularisation	transfer transferring transferred
sick sociable unexperienced blindfolded unfaithful unbelted	sick ill hospitalized hospitalised infected diagnosed
himself herself unabatedly bullhorns mcvety records gawker.com	himself ourselves herself yourself themselves myself itself

Table 2. Comparison of 9 of the 50K English word classes generated by the hirarchical clustering and the WSB clustering algorithms

To compare with other clustering algorithms, we tried the GIZA++ word clustering tool, but it could not handle the 5B word corpus (mainly due to speed). Another state-of-the-art algorithm is the hierarchical word clustering algorithm reported in (Miller et al., 2004), which is an integration of the clustering methods from (Brown et al., 1990) and (Magerman, 1995). Miller et al. (2004) showed promising results when they used it in the name tagging application. We denote this algorithm as "hierarchical clustering". We also ran this clustering on the 5B corpus to generate different numbers of classes.

We judged that the WSB clustering produced better classes, especially in terms of semantics. Table 2 shows nine classes from the "hierarchical" and "WSB" clustering that include words, "obama", "michael", "massachusetts", "lawyer", "christianity", "announce", "transfer", "sick" and "himself". As can be seen, the WSB clustering performs better in clustering synonyms together.

3.2 Clustering Chinese words

The Chinese monolingual corpus consists of 1.1B words in total: 250M words from the Chinese sentences in the 200M Chinese-English parallel corpus, and 850M words from the 4th edition of the LDC Chinese Gigaword monolingual data release[5]. We segmented the words using a 52K Chinese word lexicon[6] by using a simple left-to-right and longest-match-first algorithm. Hence, the total number of unique words was 52K.

Similarly, we ran both the "hierarchical" and "WSB" clustering algorithms on this corpus to generate different numbers of classes. We then manually compared the classes from the two clustering algorithms, and again judged that the WSB clustering produced better classes in terms of both syntax and semantics. Some clustering results are shown in Table 3, where we list 5 word classes that were also shown in Table 2 for English: "迈克尔 (michael)", "律师 (lawyer)", "输送 (transfer)", "患病 (sick)", and "基督教 (christianity)". First, one can clearly see the superiority of the WSB clustering. Second, the Chinese WSB word class for the word "基督教 (christianity)" in Table 3. shares most of the words with the corresponding English WSB word class. We see similar effects in the other 4 classes for the same root word. So the Chinese and English word classes can be treated as translations of each other. If we align up such source and target classes, they should improve the alignments for the words that belong to the classes.

Hierarchical clustering	WSB clustering
迈克尔 (michael) 马塞洛 (marcelo) 赫尔 (hull) 朱利安 (julian) 乌鲁 (uru)	迈克尔 (michael) 麦克 (mike) 丹尼尔 (daniel) 迈克 (mike)
律师 (lawyer) 调解人 (mediator) 总检察长 (prosecutor-general) 检察官 (prosecutor)	律师 (lawyer) 顾问 (counsel) 医生 (doctor) 护士 (nurse) 医师 (doctor)
输送 (transfer) 切割 (cut) 发射 (launch)	输送 (transfer) 运送 (transfer) 运载 (carry, transport)
患病 (sick) 怀孕 (pregnant) 持枪 (holding a gun)	患病 (sick) 生病 (sick) 致残 (maimed, disabled)
基督教 (christianity) 华商 (chinese business-men) SOS	基督教 (christianity) 道教 (daoism, taoism) 伊斯兰教 (islamism) 犹太教 (judaism) 天主教 (catholicism) 佛教 (budism)

Table 3. Comparison of 5 of the 25K Chinese word classes generated by the hierarchical and WSB clustering algorithms

4 Improving Word Alignments

With the word classes we proceeded to improve the word alignments. Our approach is simple and straightforward, but to the best of our knowledge we have not seen the same approach had been reported in the literature.

4.1 The procedure

Researchers have tried to generate class alignments through word alignments, such as in (Ker and Zhang, 1997, Baker et al., 2009). We trained class alignments without interaction with the word alignment training. Our procedure for improving word alignments is as follows:

1) Train word alignments on the parallel data (denoted as "regular" word alignments)
2) Replace the source and target words in the parallel data with their class names

[5] The name of this release is "LDC2009T27".

[6] We obtained this lexicon by removing infrequent words from the Chinese word lexicon released by LDC and adding the most commonly used 7K Chinese characters.

3) Train class alignments (in the same way as training word alignments in Step 1)
4) Replace the class names with the corresponding source and target words to get a new set of word alignments (denoted as "class-derived" word alignments)
5) Use both the regular and the class-derived word alignments in MT training and decoding

In Step 5), there are different ways to combine the two sets of alignments, such as the entropy approach in (Ayan and Dorr, 2006) and probability interpolating approach in (Wang et al., 2006). We concatenated the two sets of alignments together for the rule extraction – the same way as used in (Xu and Rosti, 2010).

GIZA++ also uses word classes inside its training by introducing word class dependencies in the Model4, Model5 and HMM training. Following (Och and Ney, 2003), we used the GIZA++ word clustering tool – "_mkcls" – to generate 50 source and 50 target word classes for the word alignment training. As described above, the way that we use word classes is outside the GIZA++ training. We verified that the gains from our method are additive to the gains from the use of the word classes inside GIZA++, so we used the GIZA++ word classes in all our alignment training experiments, unless specified.

4.2 Experimental results

We conducted experiments on the 165K parallel training data following the procedure described in Section 4.1. We ran experiments with different numbers of word classes, always using the same number of classes in both languages. All the experiments are shown in Table 4, where in the "word alignments" column the "+" sign means the concatenation of alignments.

The results with the hierarchical word classes show that the use of the 5K classes produced the best gain – 1.5 BLEU points, but with more classes (15K and 25K) the gain began to shrink quickly. However, the results with the WSB word classes show that more WSB classes tended to produce larger gains. The best gain was obtained when 25K classes were used – an average of 2 words per Chinese word class. We found that most of the 25K classes were single-word classes and only synonyms with high similarity were clustered together. This indicates that the WSB clustering is able to cluster synonyms together even when it produces large numbers of classes. These results verify the superior performance of the WSB clustering observed on the clustering results.

As can be seen, the use of the 25K WSB classes produced 0.6 point (=20.08-19.52) better BLEU score over the use of the 5K hierarchical classes and 2 BLEU point gain (=20.08-18.06) over the baseline – the regular word alignments. While 2 point differences might not seem large, we have recovered most of the maximum 2.3 points that could be obtained when aligning with a very large corpus (for convenience the "165K-best" system is also shown in Table 4). Considering the upper bound is 2.3 points, we would think the 0.6 and 2 point gains are significant.

Word alignments	BLEU	
	HIER	WSB
Regular	18.06	
Regular+3Kto3K	19.18	-
Regular+5Kto5K	19.52	19.65
Regular+15Kto15K	19.04	19.73
Regular+25Kto25K	18.70	20.08
Regular+30Kto30K	-	19.72
Regular+multiple	-	20.23
165K-best	20.36	

Table 4. BLEU scores from systems that use different word alignments

We also tried to concatenate multiple WSB class-derived alignments with the regular alignments. The concatenation of three sets of class-derived alignments, "5Kto5K", "15Kto15K" and "25Kto25K", produced an extra gain, as shown in the "regular+multiple" row of Table 4., which increased the total gain to 2.17 (=20.23-18.06). So, by using the WSB-class-derived word alignments we were able to recover 94% (2.17/2.3) of the losses caused by word alignment quality that was worsened because of insufficient training.

As claimed before, our major objective was to explore methods that can utilize the availability of large amounts monolingual data to improve SMT. To verify whether we benefited from the use of the large amounts of monolingual text for generating the word classes, we used only the text data from the 165K parallel corpus for generating the word classes. There are about 25K unique English words and 20K unique Chinese words in the parallel corpus. With the WSB clustering we clustered them into 2.5K, 5K and 8K classes, respectively. As expected, the clustering results were worse. The MT results with these classes are shown in Table 5.. As shown, the best gain is 0.7 BLEU points, coming from the use of

5Kto5K class-derived word alignments (18.79 vs. 18.06). Therefore, the large amounts of monolingual data produced 1.3 point extra gain (=2.0-0.7).

Alignments	BLEU
Regular	18.06
Regular+2.5Kto2.5K	18.00
Regular+5Kto5K	18.79
Regular+8Kto8K	18.70

Table 5. BLEU scores from word classes generated with WSB clustering on the 165K parallel corpus

Researchers have reported benefits from combining word alignments trained with different aligners (Baker et al., 2009, Xu and Rosti, 2010), where the benefits came from the complimentary information from the different alignments. The main reason for using class-derived word alignments was to improve alignments for infrequent words, while in the previous use of multiple aligners, if all the aligners are statistical-based, most likely none could align rare words well and thus the alignments for rare words would not be improved.

4.3 Our method vs. the GIZA++ method

Since GIZA++ also uses word class dependency features in the Model4, Model5 and HMM training, we looked into how our method interacts with the GIZA++ method. First, we re-trained the "Regular" and 'Regular+25Kto25K" systems without using word classes when running GIZA++. The performance of the 4 systems with and without using word classes in the GIZA++ training are shown in the two "165K" rows in Table 6, where "with/cls' and "wo/cls" indicate the systems trained with and without word classes in the GIZA++ training, respectively. From this set of experiments we observe: 1) the use of words in GIZA++ improved the performance by about 1 point (18.06 vs. 17.13); 2) the use of the 25Kto25K class-derived word alignments on top of the regular systems trained with and without word classes in GIZA++ produced the same gains (18.06 to 20.08 with classes vs. 17.13 to 19.08 without classes), so the gain from our method of using word classes is additive to the gain from the use of classes in GIZA++.

Second, we increased the training corpus from 165K words to 500K words and re-ran the above experiments. These experiments are shown in the "500K" rows in Table 6. As can be seen, the use of word classes in the GIZA+ training no longer helped (22.01 vs. 21.93) when the amount of training data was increased to 500K words. However, the gain from our method still held (22.01 vs. 24.07).

Training	Alignments	With/cls	Wo/cls
165K	Regular	18.06	17.13
165K	Regular+25Kto25K	20.08	19.04
500K	Regular	22.01	21.93
500K	Regular+25Kto25K	24.07	-

Table 6. MT performance with and without using word classes inside GIZA++ on different amounts of training data

5 Conclusion

We have presented a strategy that uses word class alignments to improve word alignments in the situation where only a limited parallel training data is available. We first developed the novel WSB word clustering algorithm by generalizing the word similarity metric of (Lin, 1998). This algorithm can be easily applied to any language. We observed that this new word clustering performs better than another state-of-the-art hierarchical word clustering algorithm, especially in terms of semantics. We then designed the simple but effective approach that uses word-class-derived word alignments to improve the regular word alignment quality. Our comparisons showed again that the WSB clustering is superior when used to improve the word alignments. The use of the WSB word class alignments helped recover 94% of the MT loss resulting from poor word alignment quality due to insufficient training.

Acknowledgement

This work was supported by DARPA/I2O Contract No. HR0011-06-C-0022 under the GALE program. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

Yaser Al-Onaizan, Ulrih Germann, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Daniel Marcu and Kenji Yamada. 2000, "Translating with scarce resources. In Proceedings of the National Conference on Artificial Intelligence.

Necip Fazil Ayan and Bonnie J. Dorr. 2006. "A maximum entropy approach to combining word alignments", In Proceedings of the Human Language

Technology Conference of the North American Chapter of the ACL.

Kathy Baker, Steven Bethard, Michael Blodgood, Ralf Brown, Chris Callison-Burch, Glen Coppersmith, Bonnie Dorr, Wes Filardo, Kendal Giles, et al. 2009. "Semantically Informed Machine Translation", Final report of the 2010 Summer Camp for Advanced Language Exploration (SCALE), http://web.jhu.edu/bin/u/l/HLTCOE-TechReport-002-SIMT.pdf

Jerome Bellegarda, John W. Butzberger, Yen-Lu Chow, Noah B. Coccaro, Devang Naik. 1996. "A Novel Word Clustering Algorithm Based on Latent Semantic Analysis", in Proceedings of ACSSAP 1996, Atlanta, USA.

Perter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, Robert L. Mercer. 1990. "Class-based n-gram models of natural language", Computational Linguistics.

Devid M. Magerman. 1995. "Statistical Decision-Tree Models for Parsing", in Proceedings of ACL Conference.

Maayan Geffet and Ido Dagan. 2004. "Feature Vector Quality and Distributional Similarity", in Proceedings of 20th International Conference on Computational Linguistics.

Dmitriy Genzel, Klaus Machery, Jakob Uszkoreit. 2009. "Creating a High-Quality Machine Translation System for a Low-Resource Languages: Yiddish", in Proceedings of the 20th Machine Translation Summit, 2009, Ottawa, Canada.

Petr Homola and Vladisla Kubon. 2005. "A Machine Translation System into a Minority Language", In Proceedings of the Workshop on Modern Approaches in Translation Technologies 2005, Borovets, Bulgaria

Kenichi Ichioka and Fumiyo Fukmoto. 2008. "Graph-based Clustering for Semantic Classification of Onomatopoetic Words", in Proceedings of the 3rd Text-graphs Workshop on Graph-based Algorithms for Natural Language Processing, Manchester, UK.

Sue Ker and J. Zhang. 1997. "A Class-based Approach to Word Alignment", in Computational Linguistics, Vol. 23, No. 2, pp 313-343.

Lillian Lee. 1999. Measures of Distributional Similarity. In Proceeding of the 37th Annual Meeting of the ACL, pages 25–32.

Dekang Lin. 1998. "Automatic Retrieval and Clustering of Similar Words", in Proceedings of the 17th international conference on computational linguistics. Vol. 2. Canada.

Mike Maxwell and Baden Hughes, 2006. "Frontiers in Linguistic Annotation for Lower-density Languages", in Proceedings of COLING/ACL2006 Workshop on Frontiers in Linguistically Annotated Corpora.

Scott Miller, Jethan Guinness and Alex Zamnian. 2004. "Name Tagging with Word Clusters and discriminative Training", Proceedings of HLT-NAACL 2004.

Philipe Muller, Nabil Hathout, and Bruno Gaume. 2006. "Synonym Extraction Using a Semantic Distance on a Dictionary", in Proceedings of the Workshop on TextGraphs on Graph-based Algorithms for Natural Language Processing, 2006.

Sonja Nießen and Hermann Ney. 2004. "Statistical Machine Translation with Scarce Resources Using Morphosyntactic Information", Computational Linguistics., 30(2).

Franz Och and Hermann Ney. 2003. "A systematic comparison of various statistical alignment models", Computational Linguistics, 29(1):19_51.

Franz Och. 2003. "Minimum Error Rate Training in Statistical Machine Translation", Proceedings of the 41st Annual Meeting of the ACL

Kishore Papineni, Salim Roukos, Todd Ward and Wei-jing Zhu. 2002, BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual conference of the association of computational linguistics.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. "A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model", Proceedings of the 46th Annual Meeting of the ACL.

Ravi Sinha and Rada Mihalcea. 2007. "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity", in Proceedings of the IEEE International Conference on Semantic Computing, CA, USA.

Mahesh K. Sinha. 2009. "Developing English-Urdu Machine Translation via Hindi", the 3rd Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3), MT Summit XII, Aug.26-30, 2009, Ottawa, Canada.

Haifeng Wang, Hua Wu, and Zhanyi Liu. 2006. "Word Alignment for Languages with Scarce Resources Using Bilingual Corpora of Other Language Pairs", in Proceedings of the COLING/ACL 2006.

Wei Wang and Ming Zhou. 2004. "Improving Word Alignment Models Using Structured Monolingual Corpora", in the Proceeding of EMNLP, 2004.

Julie Weeds and David Weir. 2005. "Co-oocurrence Retrival: A Flexible Framework for Lexical Distributional Similarity", in Computational Linguistics, 31(4):439–476.

Bing Xiang, Yonggang Deng, Bowen Zhou. 2010. "Diversify and Combine: Improving Word Alignment for Machine Translation on Low-Resource Languages". ACL 2008.

Jinxi Xu and Antti-Veikko I. Rosti. 2010. "Combining Unsupervised and Supervised Alignments for MT: An empirical Study", in Proceedings of EMNLP 2010.