Improving Word Alignment using Word Similarity

Theerawat Songyot
Dept of Computer Science
University of Southern California
songyot@usc.edu

David Chiang[†]
Dept of Computer Science and Engineering
University of Notre Dame
dchiang@nd.edu

Abstract

We show that semantic relationships can be used to improve word alignment, in addition to the lexical and syntactic features that are typically used. In this paper, we present a method based on a neural network to automatically derive word similarity from monolingual data. We present an extension to word alignment models that exploits word similarity. Our experiments, in both large-scale and resource-limited settings, show improvements in word alignment tasks as well as translation tasks.

1 Introduction

Word alignment is an essential step for learning translation rules in statistical machine translation. The task is to find word-level translation correspondences in parallel text. Formally, given a source sentence $\mathbf{e}$ consisting of words $e_1, e_2, \ldots, e_l$ and a target sentence $\mathbf{f}$ consisting of words $f_1, f_2, \ldots, f_m$, we want to infer an alignment $\mathbf{a}$, a sequence of indices $a_1, a_2, \ldots, a_m$ which indicates, for each target word f_i, the corresponding source word e_{a_i} or a null word. Machine translation systems, including state-of-the-art systems, then use the word-aligned corpus to extract translation rules.

The most widely used methods, the IBM models (Brown et al., 1993) and HMM (Vogel et al., 1996), define a probability distribution $p(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$ that models how each target word f_i is generated from a source word e_{a_i} with respect to an alignment $\mathbf{a}$. The models, however, tend to misalign low-frequency words as they have insufficient training samples. The problem can get worse in low-resource languages. Two branches of research have tried to alleviate the problem. The first branch relies solely on the parallel data; however, additional assumptions about the data are required. This includes, but is not limited to, applying prior distributions (Mermer and Saraçlar, 2011; Vaswani et al., 2012) or smoothing techniques (Zhang and Chiang, 2014). The other branch uses information learned from monolingual data, which is generally easier to acquire than parallel data. Previous work in this branch mostly involves applying syntactic constraints (Yamada and Knight, 2001; Cherry and Lin, 2006; Wang and Zong, 2013) and syntactic features (Toutanova et al., 2002) into the models. The use of syntactic relationships can, however, be limited between historically unrelated language pairs.

Our motivation lies in the fact that a meaningful sentence is not merely a grammatically structured sentence; its semantics can provide insightful information for the task. For example, suppose that the models are uncertain about aligning e to f. If the models are informed that e is semantically related to e', f is semantically related to f', and f' is a translation of e', it should intuitively increase the probability that f is a translation of e. Our work focuses on using such a semantic relationship, in particular, word similarity, to improve word alignments.

In this paper, we propose a method to learn similar words from monolingual data (Section 2) and an extension to word alignment models in which word similarity can be incorporated (Section 3). We demonstrate its application in word alignment and translation (Section 4) and then briefly discuss the novelty of our work in comparison to other methods (Section 5).

2 Learning word similarity

Given a word w, we want to learn a word similarity model $p(w' \mid w)$ of what words w' might be used in place of w. Word similarity can be used to improve word alignment, as in this pa-

[†]Most of the work reported here was performed while the second author was at the University of Southern California.

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1840–1845,
October 25-29, 2014, Doha, Qatar.

per, but can potentially be useful for other natural language processing tasks as well. Such a model might be obtained from a monolingual thesaurus, in which humans manually provide subjective evaluation for word similarity probabilities, but an automatic method would be preferable. In this section, we present a direct formulation of the word similarity model, which can automatically be trained from monolingual data, and then consider a more practical variant, which we adopt in our experiments.

2.1 Model

Given an arbitrary word type w, we define a word similarity model $p(w' \mid w)$ for all word types w' in the vocabulary V as

$$p(w' \mid w) = \sum_c p(c \mid w)\, p(w' \mid c)$$

where c is a word context represented by a sequence $w_1, w_2, \ldots, w_{2n}$ consisting of n word tokens on the left and n word tokens on the right of w, excluding w. The submodel $p(c \mid w)$ can be a categorical distribution. However, modeling the word context model, $p(w' \mid c)$, as a categorical distribution would cause severe overfitting, because the number of all possible contexts is $|V|^{2n}$, which is exponential in the length of the context. We therefore parameterize it using a feedforward neural network as shown in Figure 1, since the structure has been shown to be effective for language modeling (Bengio et al., 2006; Vaswani et al., 2013). The input to the network is a one-hot representation of each word in c, where the special symbols <s>, </s>, <unk> are reserved for sentence beginning, sentence ending, and words not in the vocabulary. There is an output node for each $w' \in V$, whose activation is $p(w' \mid c)$. Following Bengio et al. (2006), the network uses a shared linear projection matrix to the input embedding layer, which allows information sharing among the context words and also substantially reduces the number of parameters. The input embedding layer has a dimensionality of 150 for each input word. The network uses two hidden layers with 1,000 and 150 rectified linear units, respectively, and a softmax output layer. We arbitrarily use $n = 5$ throughout this paper.

2.2 Training

We extract training data by either collecting or sampling the target words $w \in V$ and their word contexts from monolingual data. The submodel $p(c \mid w)$ can be independently trained easily by maximum likelihood estimation, while the word context model $p(w' \mid c)$ may be difficult to train at scale. We follow previous work (Mnih and Teh, 2012; Vaswani et al., 2013) in adopting noise-contrastive estimation (Gutmann and Hyvärinen, 2010), a fast and simple training algorithm that scales independently of the vocabulary size.

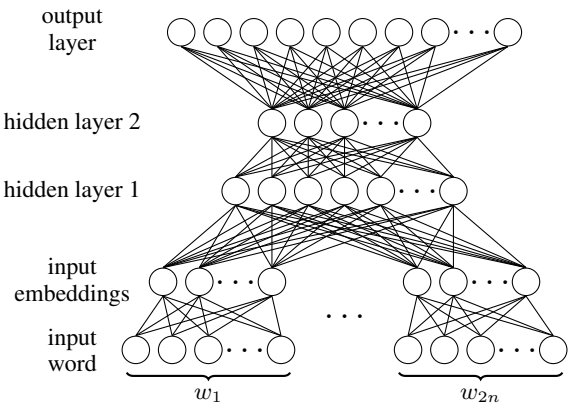

Figure 1: The structure of the word context model

2.3 Model variants

The above formulation of the word similarity model can be interpreted as a mixture model in which w' is similar to w if any of the context probabilities agrees. However, to guard against false positives, we can alternatively reformulate it as a product of experts (Hinton, 1999),

$$p(w' \mid w) = \frac{1}{Z(w)} \exp \sum_c p(c \mid w) \log p(w' \mid c)$$

where $Z(w)$ is a normalization constant. Under this model, w' is similar to w if *all* of the context probabilities agree. Both methods produce reasonably good word similarity; however, in practice, the latter performs better.

Since most of the $p(w' \mid w)$ will be close to zero, for computational efficiency, we can select the k most similar words and renormalize the probabilities. Table 1 shows some examples learned from the 402M-word Xinhua portion of the English Gigaword corpus (LDC2007T07), using a vocabulary V of the 30,000 most frequent words. We set $k = 5$ for illustration purposes.

3 Word alignment model

In this section, we present our word alignment models by extending the standard IBM models.

$p(w' \mid \text{country})$		$p(w' \mid \text{region})$		$p(w' \mid \text{area})$	
country	0.8363	region	0.8338	area	0.8551
region	0.0558	area	0.0760	region	0.0524
nation	0.0522	country	0.0524	zone	0.0338
world	0.0282	province	0.0195	city	0.0326
city	0.0273	city	0.0181	areas	0.0258

Table 1: Examples of word similarity

The method can easily be applied to other related models, for example, the log-linear reparameterization of Model 2 by Dyer et al. (2013). Basically, all the IBM models involve modeling lexical translation probabilities $p(f \mid e)$ which are parameterized as categorical distributions. IBM Model 1, for instance, is defined as

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \propto \prod_{i=1}^{m} p(f_i \mid e_{a_i}) = \prod_{i=1}^{m} t(f_i \mid e_{a_i})$$

where each $t(f \mid e)$ denotes the model parameters directly corresponding to $p(f \mid e)$. Models 2–5 and the HMM-based model introduce additional components in order to capture word ordering and word fertility. However, they have $p(f \mid e)$ in common.

3.1 Model

To incorporate word similarity in word alignment models, we redefine the lexical translation probabilities as

$$p(f \mid e) = \sum_{e', f'} p(e' \mid e)\, t(f' \mid e')\, p(f \mid f')$$

for all f, e, including words not in the vocabulary. While the factor $p(e' \mid e)$ can be directly computed by the word similarity model, the factor $p(f \mid f')$ can be problematic because it vanishes for f out of vocabulary. One possible solution would be to use Bayes' rule

$$p(f \mid f') = \frac{p(f' \mid f)\, p(f)}{p(f')}$$

where $p(f' \mid f)$ is computed by the word similarity model. However, we find that this is prone to numerical instability and other complications. In our experiments, we tried the simpler assumption that $p(f \mid f') \approx p(f' \mid f)$, with the rationale that both probabilities are measures of word similarity, which is intuitively a symmetric relation. We also compared the performance of both methods. Table 2 shows that this simple solution works as well as the more exact method of using Bayes' rule. We describe the experiment details in Section 4.

Model	F1	BLEU	
		Test 1	Test 2
Chinese-English			
Bayes' rule	75.7	30.0	27.0
Symmetry assumption	75.3	29.9	27.0
Arabic-English			
Bayes' rule	70.4	37.9	36.7
Symmetry assumption	69.5	38.2	36.8

Table 2: Assuming that word similarity is symmetric, i.e. $p(f \mid f') \approx p(f' \mid f)$, works as well as computing $p(f \mid f')$ using Bayes' rule.

3.2 Re-estimating word similarity

Depending on the quality of word similarity and the distribution of words in the parallel data, applying word similarity directly to the model could lead to an undesirable effect where similar but not interchangeable words rank in the top of the translation probabilities. On the other hand, if we set

$$p(e' \mid e) = \mathbb{1}[e' = e]$$
$$p(f' \mid f) = \mathbb{1}[f' = f]$$

where $\mathbb{1}$ denotes the indicator function, the model reduces to the standard IBM models. To get the best of both worlds, we smooth the two models together so that we rely more on word similarity for rare words and less for frequent words

$$\tilde{p}(w' \mid w) = \frac{\text{count}(w)\mathbb{1}[w' = w] + \alpha\, p(w' \mid w)}{\text{count}(w) + \alpha}$$

This can be thought of as similar to Witten-Bell smoothing, or adding α pseudocounts distributed according to our $p(w' \mid w)$. The hyperparameter α controls how much influence our word similarity model has. We investigated the effect of α by varying this hyperparameter in our word alignment experiments whose details are described in Section 4. Figure 2 shows that performance of the model, as measured by F1 score, is rather insensitive to the choice of α. We used a value of 40 in our experiments.

3.3 Training

Our word alignment models can be trained in the same way as the IBM models using the Expectation Maximization (EM) algorithm to maximize the likelihood of the parallel data. Our extension only introduces an additional time complexity on the order of $\mathcal{O}(k^2)$ on top of the base models, where k is the number of word types used to estimate the full-vocabulary word similarity models.

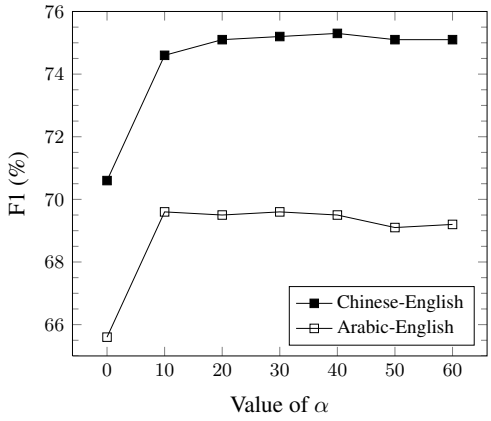

Figure 2: Alignment F1 is fairly insensitive to α over a large range of values

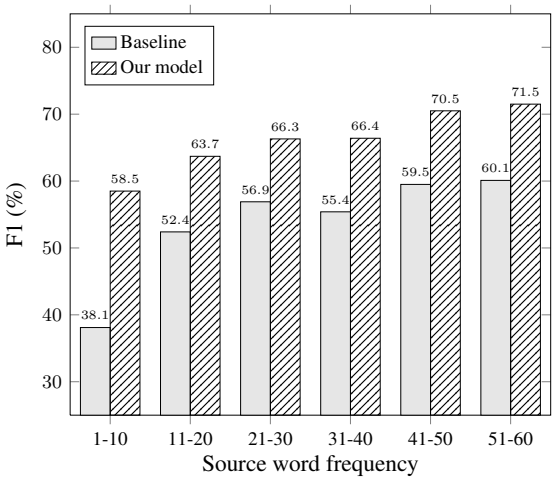

Figure 3: F1 scores for words binned by frequency. Our model gives the largest improvements for the lowest-frequency words.

The larger the value of k is, the closer to the full-vocabulary models our estimations are. In practice, a small value of k seems to be effective since $p(w' \mid w)$ is negligibly small for most w'.

4 Experiments

4.1 Alignment experiments

We conducted word alignment experiments on 2 language pairs: Chinese-English and Arabic-English. For Chinese-English, we used 9.5M+12.3M words of parallel text from the NIST 2009 constrained task[1] and evaluated on 39.6k+50.9k words of hand-aligned data (LDC2010E63, LDC2010E37). For Arabic-English, we used 4.2M+5.4M words of parallel text from the NIST 2009 constrained task[2] and evaluated on 10.7k+15.1k words of hand-aligned data (LDC2006E86). To demonstrate performance under resource-limited settings, we additionally experimented on only the first eighth of the full data, specifically, 1.2M+1.6M words for Chinese-English and 1.0M+1.4M words for Arabic-English. We trained word similarity models on the Xinhua portions of English Gigaword (LDC2007T07), Chinese Gigaword (LDC2007T38), and Arabic Gigaword (LDC2011T1), which are 402M, 323M, and 125M words, respectively. The vocabulary V was the 30,000 most frequent words from each corpus and the $k = 10$ most similar words were used.

We modified GIZA++ (Och and Ney, 2003) to incorporate word similarity. For all experiments, we used the default configuration of GIZA++: 5 iterations each of IBM Model 1, 2, HMM, 3 and 4. We aligned the parallel texts in both forward and backward directions and symmetrized them using grow-diag-final-and (Koehn et al., 2005). We evaluated alignment quality using precision, recall, and F1.

The results in Table 3 suggest that our modeling approach produces better word alignments. We found that our models not only learned smoother translation models for low frequency words but also ranked the conditional probabilities more accurately with respect to the correct translations. To illustrate this, we categorized the alignment links from the Chinese-English low-resource experiment into bins with respect to the English source word frequency and individually evaluated them. As shown in Figure 3, the gain for low frequency words is particularly large.

4.2 Translation experiments

We also ran end-to-end translation experiments. For both languages, we used subsets of the NIST 2004 and 2006 test sets as development data. We used two different data sets as test data: different subsets of the NIST 2004 and 2006 test sets (called Test 1) and the NIST 2008 test sets (called Test 2). We trained a 5-gram language model on the Xinhua portion of English Gigaword (LDC2007T07). We used the Moses toolkit (Koehn et al., 2007) to

[1]Catalog numbers: LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E86, LDC2006E92, and LDC2006E93.

[2]Excluding: United Nations proceedings (LDC2004E13), ISI Automatically Extracted Parallel Text (LDC2007E08), and Ummah newswire text (LDC2004T18)

Model	Precision	Recall	F1	BLEU		METEOR	
				Test 1	Test 2	Test 1	Test 2
Chinese-English							
Baseline	65.2	76.9	70.6	29.4	26.7	29.7	28.5
Our model	71.4	79.7	75.3	29.9	27.0	30.0	28.8
Baseline (resource-limited)	56.1	68.1	61.5	23.6	20.3	26.0	24.4
Our model (resource-limited)	66.5	74.4	70.2	24.7	21.6	26.8	25.6
Arabic-English							
Baseline	56.1	79.0	65.6	37.7	36.2	31.1	30.9
Our model	60.0	82.4	69.5	38.2	36.8	31.6	31.4
Baseline (resource-limited)	56.7	76.1	65.0	34.1	33.0	27.9	27.7
Our model (resource-limited)	59.4	80.7	68.4	35.0	33.8	28.7	28.6

Table 3: Experimental results. Our model improves alignments and translations on both language pairs.

build a hierarchical phrase-based translation system (Chiang, 2007) trained using MIRA (Chiang, 2012). Then, we evaluated the translation quality using BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014), and performed significance testing using bootstrap resampling (Koehn, 2004) with 1,000 samples.

Under the resource-limited settings, our methods consistently show 1.1–1.3 BLEU (0.8–1.2 METEOR) improvements on Chinese-English and 0.8–0.9 BLEU (0.8–0.9 METEOR) improvements on Arabic-English, as shown in Table 3. These improvements are statistically significant ($p < 0.01$). On the full data, our method improves Chinese-English translation by 0.3–0.5 BLEU (0.3 METEOR), which is unfortunately not statistically significant, and Arabic-English translation by 0.5–0.6 BLEU (0.5 METEOR), which is statistically significant ($p < 0.01$).

5 Related work

Most previous work on word alignment problems uses morphosyntactic-semantic features, for example, word stems, content words, orthography (De Gispert et al., 2006; Hermjakob, 2009). A variety of log-linear models have been proposed to incorporate these features (Dyer et al., 2011; Berg-Kirkpatrick et al., 2010). These approaches usually require numerical optimization for discriminative training as well as language-specific engineering and may limit their applications to morphologically rich languages.

A more semantic approach resorts to training word alignments on semantic word classes (Ma et al., 2011). However, the resulting alignments are only used to supplement the word alignments learned on lexical words. To our knowledge, our work, which directly incorporates semantic relationships in word alignment models, is novel.

6 Conclusion

We have presented methods to extract word similarity from monolingual data and apply it to word alignment models. Our method can learn similar words and word similarity probabilities, which can be used inside any probability model and in many natural language processing tasks. We have demonstrated its effectiveness in statistical machine translation. The enhanced models can significantly improve alignment quality as well as translation quality.

Acknowledgments

We express our appreciation to Ashish Vaswani for his advice and assistance. We also thank Hui Zhang, Tomer Levinboim, Qing Dou, Aliya Deri for helpful discussions and the anonymous reviewers for their insightful critiques. This research was supported in part by DOI/IBC grant D12AP00225 and a Google Research Award to Chiang.

References

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of HLT NAACL*, pages 582–590.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation:

Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of COLING/ACL*, pages 105–112.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research*, 13(1):1159–1187.

Adrià De Gispert, Deepa Gupta, Maja Popović, Patrik Lambert, Jose B. Mariño, Marcello Federico, Hermann Ney, and Rafael Banchs. 2006. Improving statistical word alignments with morpho-syntactic transformations. In *Advances in Natural Language Processing*, pages 368–379. Springer.

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

Chris Dyer, Jonathan Clark, Alon Lavie, and Noah A. Smith. 2011. Unsupervised word alignment with arbitrary features. In *Proceedings of ACL: HLT*, volume 1, pages 409–419.

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

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics (AI-STATS)*, pages 297–304.

Ulf Hermjakob. 2009. Improved word alignment with statistics and linguistic heuristics. In *Proceedings of EMNLP*, volume 1, pages 229–237.

Geoffrey E. Hinton. 1999. Products of experts. In *International Conference on Artificial Neural Networks*, volume 1, pages 1–6.

Philipp Koehn, Amittai Axelrod, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL: Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

Jeff Ma, Spyros Matsoukas, and Richard Schwartz. 2011. Improving low-resource statistical machine translation with a novel semantic word clustering algorithm. In *Proceedings of MT Summit*.

Coşkun Mermer and Murat Saraçlar. 2011. Bayesian word alignment for statistical machine translation. In *Proceedings of ACL: HLT*, volume 2, pages 182–187.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of ICML*.

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

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

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. 2002. Extensions to HMM-based statistical word alignment models. In *Proceedings of EMNLP*, pages 87–94.

Ashish Vaswani, Liang Huang, and David Chiang. 2012. Smaller alignment models for better translations: unsupervised word alignment with the ℓ_0-norm. In *Proceedings of ACL*, volume 1, pages 311–319.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of EMNLP*.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING*, volume 2, pages 836–841.

Zhiguo Wang and Chengqing Zong. 2013. Large-scale word alignment using soft dependency cohesion constraints. *Transactions of the Association for Computational Linguistics*, 1(6):291–300.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL*, pages 523–530.

Hui Zhang and David Chiang. 2014. Kneser-Ney smoothing on expected counts. In *Proceedings of ACL*.