A Beam-Search Extraction Algorithm for Comparable Data

Christoph Tillmann
IBM T.J. Watson Research Center
Yorktown Heights, N.Y. 10598
ctill@us.ibm.com

Abstract

This paper extends previous work on extracting parallel sentence pairs from comparable data (Munteanu and Marcu, 2005). For a given source sentence S, a maximum entropy (ME) classifier is applied to a large set of candidate target translations . A beam-search algorithm is used to abandon target sentences as non-parallel early on during classification if they fall outside the beam. This way, our novel algorithm avoids any document-level prefiltering step. The algorithm increases the number of extracted parallel sentence pairs significantly, which leads to a BLEU improvement of about 1 % on our Spanish-English data.

1 Introduction

The paper presents a novel algorithm for extracting parallel sentence pairs from comparable monolingual news data. We select source-target sentence pairs (S, T) based on a ME classifier (Munteanu and Marcu, 2005). Because the set of target sentences T considered can be huge, previous work (Fung and Cheung, 2004; Resnik and Smith, 2003; Snover et al., 2008; Munteanu and Marcu, 2005) pre-selects target sentences T at the document level . We have re-implemented a particular filtering scheme based on BM25 (Quirk et al., 2007; Utiyama and Isahara, 2003; Robertson et al., 1995). In this paper, we demonstrate a different strategy . We compute the ME score incrementally at the word level and apply a beam-search algorithm to a large number of sentences. We abandon target sentences early on during classification if they fall outside the beam. For comparison purposes, we run our novel extraction algorithm with and without the document-level prefiltering step. The results in Section 4 show that the number of extracted sentence pairs is more than doubled which also leads to an increase in BLEU by about 1 % on the Spanish-English data.

The classification probability is defined as follows:

$$p(c|S,T) \quad = \quad \frac{exp(\ w^T \cdot f(c,S,T)\ )}{Z(S,T)}\ , \quad (1)$$

where $S = s_1^J$ is a source sentence of length J and $T = t_1^I$ is a target sentence of length I. $c \in \{0,1\}$ is a binary variable . $p(c|S,T) \in [0,1]$ is a probability where a value $p(c = 1|S,T)$ close to 1.0 indicates that S and T are translations of each other. $w \in \mathbb{R}^n$ is a weight vector obtained during training. $f(c,S,T)$ is a feature vector where the features are co-indexed with respect to the alignment variable c. Finally, $Z(S,T)$ is an appropriately chosen normalization constant.

Section 2 summarizes the use of the binary classifier. Section 3 presents the beam-search algorithm. In Section 4, we show experimental results. Finally, Section 5 discusses the novel algorithm.

2 Classifier Training

The classifier in Eq. 1 is based on several real-valued feature functions f_i . Their computation is based on the so-called IBM Model-1 (Brown et al., 1993). The Model-1 is trained on some parallel data available for a language pair, i.e. the data used to train the baseline systems in Section 4. $p(s|T)$ is the Model-1 probability assigned to a source word s given the target sentence T , $p(t|S)$ is defined accordingly. $p(s|t)$ and $p(t|s)$ are word translation probabilities obtained by two parallel Model-1 training steps on the same data, but swapping the role of source and target language. To compute these values efficiently, the implementation techniques in (Tillmann and Xu, 2009) are used. **Coverage** and **fertility** features are defined based on the Model-1 Viterbi alignment: a source

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 225–228,
Suntec, Singapore, 4 August 2009. ©2009 ACL and AFNLP

word s is said to be **covered** if there is a target word $t \in T$ such that its probability is above a threshold ϵ: $p(s|t) > \epsilon$. We define the **fertility** of a source word s as the number of target words $t \in T$ for which $p(s|t) > \epsilon$. Target word coverage and fertility are defined accordingly. A large number of 'uncovered' source and target positions as well as a large number of high fertility words indicate non-parallelism. We use the following $N = 7$ features: 1,2) lexical Model-1 weighting: $\sum_s -log(\ p(s|T)\ )$ and $\sum_t -log(\ p(t|S)\ )$, 3,4) number of uncovered source and target positions, 5,6) sum of source and target fertilities, 7) number of covered source and target positions . These features are defined in a way that they can be computed incrementally at the word level. Some thresholding is applied, e.g. a sequence of uncovered positions has to be at least 3 positions long to generate a non-zero feature value . In the feature vector $f(c, S, T)$, each feature f_i occurs potentially twice, once for each class $c \in \{0, 1\}$. For the feature vector $f(c = 1, S, T)$, all the feature values corresponding to class $c = 0$ are set to 0, and vice versa. This particular way of defining the feature vector is needed for the search in Section 3: the contribution of the 'negative' features for $c = 0$ is only computed when Eq. 1 is evaluated for the highest scoring final hypothesis in the beam. To train the classifier, we have manually annotated a collection of 524 sentence pairs . A sentence pair is considered parallel if at least 75 % of source and target words have a corresponding translation in the other sentence, otherwise it is labeled as non-parallel. A weight vector $w \in \mathbb{R}^{2*N}$ is trained with respect to classification accuracy using the on-line maxent training algorithm in (Tillmann and Zhang, 2007).

3 Beam Search Algorithm

We process the comparable data at the sentence level: sentences are indexed based on their publication date. For each source sentence S, a matching score is computed over all the target sentences $T_m \in \Theta$ that have a publication date which differs less than 7 days from the publication date of the source sentence [1]. We are aiming at finding the $\hat{T}$ with the highest probability $p(c = 1|S, \hat{T})$, but we cannot compute that probability for all sentence pairs (S, T_m) since $|\Theta|$ can be in tens of thousands of sentences . Instead, we use a beam-search algorithm to search for the sentence pair $(S, \hat{T})$ with the highest matching score $w^T \cdot f(1, S, \hat{T})$ [2]. The 'light-weight' features defined in Section 2 are such that the matching score can be computed incrementally while processing the source and target sentence positions in some order. To that end, we maintain a stack of matching hypotheses for each source position j. Each hypothesis is assigned a partial matching score based on the source and target positions processed so far. Whenever a partial matching score is low compared to partial matching scores of other target sentence candidates, that translation pair can be discarded by carrying out a beam-search pruning step. The search is organized in a single left-to-right run over the source positions $1 \leq j \leq J$ and all active partial hypotheses match the same portion of that source sentence. There is at most a single active hypothesis for each different target sentence T_i, and search states are defined as follows:

$$[\ m\ ,\ j\ ,\ u_j\ ,\ u_i\ ;\ d\ ]\ .$$

Here, $m \in \{1, \cdots, |\Theta|\}$ is a target sentence index. j is a position in the source sentence, u_j and u_i are the number of uncovered source and target positions to the left of source position j and target position i (coverage computation is explained above), and d is the partial matching score . The target position i corresponding to the source position j is computed deterministically as follows:

$$i \;=\; \lceil I \cdot \frac{j}{J} \rceil\ , \tag{2}$$

where the sentence lengths I and J are known for a sentence pair (S, T). Covering an additional source position leads to covering additional target positions as well, and source and target features are computed accordingly. The search is initialized by adding a single hypothesis for each target sentence $T_m \in \Theta$ to the stack for $j = 1$:

$$[\ m\ ,\ j = 1\ ,\ u_j = 0\ ,\ u_i = 0\ ;\ 0\ ]\ .$$

During the left-to-right search , state transitions of the following type occur:

$$[\ m\ ,\ j\ ,\ u_j\ ,\ u_i\ ;\ d\ ] \rightarrow \\ [\ m\ ,\ j+1\ ,\ u_j^{'}\ ,\ u_i^{'}\ ;\ d'\ ]\ ,$$

[1] In addition, the sentence length filter in (Munteanu and Marcu, 2005) is used: the length ratio $max(J, I)/min(J, I)$ of source and target sentence has to be smaller than 2.

[2] This is similar to standard phrase-based SMT decoding, where a set of real-valued features is used and any sentence-level normalization is ignored during decoding. We assume the effect of this approximation to be small.

where the partial score is updated as: $d' = d + w^T \cdot f(1, j, i)$. Here, $f(1, j, i)$ is a partial feature vector computed for all the additional source and target positions processed in the last extension step. The number of uncovered source and target positions u' is updated as well. The beam-search algorithm is carried out until all source positions j have been processed. We extract the highest scoring partial hypothesis from the final stack $j = J$. For that hypothesis, we compute a global feature vector $f(1, S, T)$ by adding all the local $f(1, j, i)$'s component-wise. The 'negative' feature vector $f(0, S, T)$ is computed from $f(1, S, T)$ by copying its feature values. We then use Eq. 1 to compute the probability $p(1|S, T)$ and apply a threshold of $\theta = 0.75$ to extract parallel sentence pairs. We have adjusted beam-search pruning techniques taken from regular SMT decoding (Tillmann et al., 1997; Koehn, 2004) to reduce the number of hypotheses after each extension step. Currently, only histogram pruning is employed to reduce the number of hypotheses in each stack.

The resulting beam-search algorithm is similar to a monotone decoder for SMT: rather then incrementally generating a target translation, the decoder is used to select entire target sentences out of a pre-defined list. That way, our beam search algorithm is similar to algorithms in large-scale speech recognition (Ney, 1984; Vintsyuk, 1971), where an acoustic signal is matched to a pre-assigned list of words in the recognizer vocabulary.

4 Experiments

The parallel sentence extraction algorithm presented in this paper is tested in detail on all of the large-scale Spanish-English Gigaword data (Graff, 2006; Graff, 2007) as well as on some smaller Portuguese-English news data . For the Spanish-English data , matching sentence pairs come from the same news feed. Table 1 shows the size of the comparable data, and Table 2 shows the effect of including the additional sentence pairs into the training of a phrase-based SMT system. Here, both languages use a test set with a single reference. The test data comes from Spanish and Portuguese news web pages that have been translated into English. Including about 1.35 million sentence pairs extracted from the Gigaword data, we obtain a statistically significant improvement from 42.3 to 45.7 in BLEU. The baseline system has been trained on about 1.8 million sentence pairs from Europarl and FBIS parallel data. We also present results for a Portuguese-English system: the baseline has been trained on Europarl and JRC data. Parallel sentence pairs are extracted from comparable news data published in 2006. For this data, no document-level information was available. To gauge the effect of the document-level pre-filtering step, we have re-implemented an IR technique based on BM25 (Robertson et al., 1995). This type of pre-filtering has also been used in (Quirk et al., 2007; Utiyama and Isahara, 2003). We split the Spanish data into documents. Each Spanish document is translated into a bag of English words using Model-1 lexicon probabilities trained on the baseline data. Each of these English bag-of-words is then issued as a query against all the English documents that have been published within a 7 day window of the source document. We select the 20 highest scoring English documents for each source document . These 20 documents provide a restricted set of target sentence candidates. The sentence-level beam-search algorithm without the document-level filtering step searches through close to 1 trillion sentence pairs. For the data obtained by the BM25-based filtering step, we still use the same beam-search algorithm but on a much smaller candidate set of only 25.4 billion sentence pairs. The probability selection threshold θ is determined on some development set in terms of precision and recall (based on the definitions in (Munteanu and Marcu, 2005)). The classifier obtains an F-measure classifications performance of about 85 %. The BM25 filtering step leads to a significantly more complex processing pipeline since sentences have to be indexed with respect to document boundaries and publication date. The document-level pre-filtering reduces the overall processing time by about 40 % (from 4 to 2.5 days on a 100-CPU cluster). However, the exhaustive sentence-level search improves the BLEU score by about 1 % on the Spanish-English data.

Table 1: Corpus statistics for comparable data.

	Spanish	English
Sentences	19.4 million	47.9 million
Words	601.5 million	1.36 billion
	Portuguese	English
Sentences	366.0 thousand	5.3 million
Words	11.6 million	171.1 million

Table 2: Spanish-English and Portuguese-English extraction results. Extraction threshold is $\theta = 0.75$ for both language pairs. # cands reports the size of the overall search space in terms of sentence pairs processed .

Data Source	# cands	# pairs	Bleu
Baseline	-	$1.826\ M$	42.3
+ Giga	$999.3\ B$	$1.357\ M$	45.7
+ Giga (BM25)	$25.4\ B$	$0.609\ M$	44.8
Baseline	-	$2.222\ M$	45.3
+ News Data 2006	$77.8\ B$	$56\ K$	47.2

5 Future Work and Discussion

In this paper, we have presented a novel beam-search algorithm to extract sentence pairs from comparable data . It can avoid any pre-filtering at the document level (Resnik and Smith, 2003; Snover et al., 2008; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005; Fung and Cheung, 2004). The novel algorithm is successfully evaluated on news data for two language pairs. A related approach that also avoids any document-level pre-filtering has been presented in (Tillmann and Xu, 2009). The efficient implementation techniques in that paper are extended for the ME classifier and beam search algorithm in the current paper, i.e. feature function values are cached along with Model-1 probabilities.

The search-driven extraction algorithm presented in this paper might also be applicable to other NLP extraction task, e.g. named entity extraction. Rather then employing a cascade of filtering steps, a one-stage search with a specially adopted feature set and search space organization might be carried out . Such a search-driven approach makes less assumptions about the data and may increase the number of extracted entities, i.e. increase recall.

Acknowledgments

We would like to thanks the anonymous reviewers for their valuable remarks.

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *CL*, 19(2):263–311.

Pascale Fung and Percy Cheung. 2004. Mining Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM. In *Proc, of EMNLP 2004*, pages 57–63, Barcelona, Spain, July.

Dave Graff. 2006. *LDC2006T12: Spanish Gigaword Corpus First Edition.* LDC.

Dave Graff. 2007. *LDC2007T07: English Gigaword Corpus Third Edition.* LDC.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *Proceedings of AMTA'04*, Washington DC, September-October.

Dragos S. Munteanu and Daniel Marcu. 2005. Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *CL*, 31(4):477–504.

H. Ney. 1984. The Use of a One-stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):263–271.

Chris Quirk, Raghavendra Udupa, and Arul Menezes. 2007. Generative Models of Noisy Translations with Applications to Parallel Fragment Extraction. In *Proc. of the MT Summit XI*, pages 321–327, Copenhagen,Demark, September.

Philip Resnik and Noah Smith. 2003. The Web as Parallel Corpus. *CL*, 29(3):349–380.

S E Robertson, S Walker, M M Beaulieu, and M Gatford. 1995. Okapi at TREC-4. In *Proc. of the 4th Text Retrieval Conference (TREC-4)*, pages 73–96.

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and Translation Model Adaptation using Comparable Corpora. In *Proc. of EMNLP08*, pages 856–865, Honolulu, Hawaii, October.

Christoph Tillmann and Jian-Ming Xu. 2009. A Simple Sentence-Level Extraction Algorithm for Comparable Data. In *Companion Vol. of NAACL HLT 09*, pages 93–96, Boulder, Colorado, June.

Christoph Tillmann and Tong Zhang. 2007. A Block Bigram Prediction Model for Statistical Machine Translation. *ACM-TSLP*, 4(6):1–31, July.

Christoph Tillmann, Stefan Vogel, Hermann Ney, and Alex Zubiaga. 1997. A DP-based Search Using Monotone Alignments in Statistical Translation. In *Proc. of ACL 97*, pages 289–296, Madrid,Spain, July.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable Measures for Aligning Japanese-English News Articles and Sentences. In *Proc. of ACL03*, pages 72–79, Sapporo, Japan, July.

T.K. Vintsyuk. 1971. Element-Wise Recognition of Continuous Speech Consisting of Words From a Specified Vocabulary. *Cybernetics (Kibernetica)*, (2):133–143, March-April.