A Maximum Entropy Word Aligner for Arabic-English Machine Translation

Abraham Ittycheriah and Salim Roukos
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
{abei,roukos}@us.ibm.com

Abstract

This paper presents a maximum entropy word alignment algorithm for Arabic-English based on *supervised* training data. We demonstrate that it is feasible to create training material for problems in machine translation and that a mixture of supervised and unsupervised methods yields superior performance. The probabilistic model used in the alignment directly models the link decisions. Significant improvement over traditional word alignment techniques is shown as well as improvement on several machine translation tests. Performance of the algorithm is contrasted with human annotation performance.

1 Introduction

Machine translation takes a source sequence,

$$S = [s_1\ s_2\ \ldots\ s_K]$$

and generates a target sequence,

$$T = [t_1\ t_2\ \ldots\ t_M]$$

that renders the meaning of the source sequence into the target sequence. Typically, algorithms operate on sentences. In the most general setup, one or more source words can generate 0, 1 or more target words. Current state of the art machine translation systems (Och, 2003) use phrasal (n-gram) features extracted automatically from parallel corpora. These phrases are extracted using word alignment algorithms that are trained on parallel corpora. Phrases, or phrasal features, represent a mapping of source sequences into a target sequences which are typically a few words long.

In this paper, we investigate the feasibility of training alignment algorithms based on supervised alignment data. Although there is a modest cost associated with annotating data, we show that a reduction of 40% relative in alignment error (AER) is possible over the GIZA++ aligner (Och and Ney, 2003).

Although there are a number of other applications for word alignment, for example in creating bilingual dictionaries, the primary application continues to be as a component in a machine translation system. We test our aligner on several machine translation tests and show encouraging improvements.

2 Related Work

Most of the prior work on word alignments has been done on parallel corpora where the alignment at the sentence level is also done automatically. The IBM models 1-5 (Brown et al., 1993) produce word alignments with increasing algorithmic complexity and performance. These IBM models and more recent refinements (Moore, 2004) as well as algorithms that bootstrap from these models like the HMM algorithm described in (Vogel et al., 1996) are unsupervised algorithms.

The relative success of these automatic techniques together with the human annotation cost has delayed the collection of supervised word-aligned corpora for more than a decade.

(Cherry and Lin, 2003) recently proposed a direct alignment formulation and state that it would be straightforward to estimate the parameters given a supervised alignment corpus. In this paper, we extend their work and show that with a small amount of annotated data, together with a modeling strategy and search algorithm yield significant gains in alignment F-measure.

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 89–96, Vancouver, October 2005.

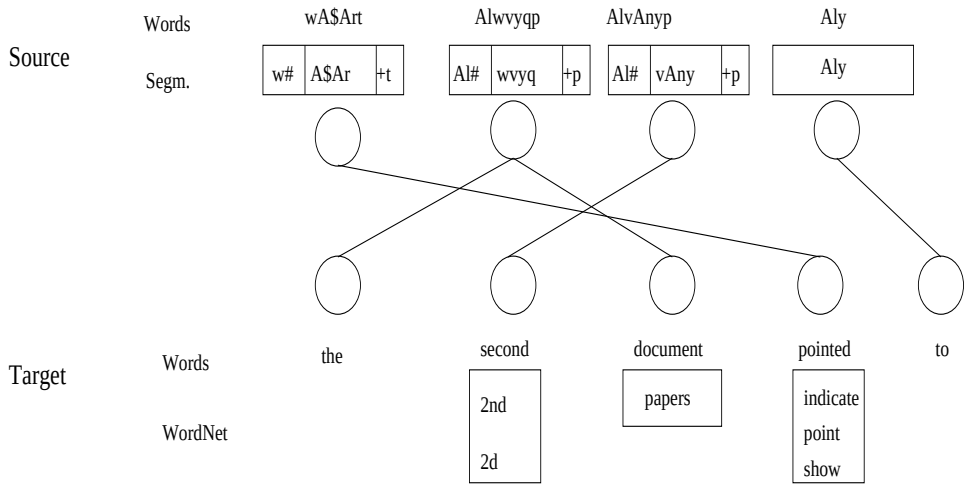

Figure 1: Alignment example.

3 Algorithm

In order to describe the algorithm, we will need to first describe the direct link model. Figure 1 shows two sequences where the top sequence is considered the source sequence and the bottom sequence the target sequence. Each sequence can have auxilliary information such as Arabic segmentation or English WordNet (Miller, 1990) information as shown. Given the source and target sequences, there are a number of different ways to link each target word to a source word. Each target word has a link l_i which indicates which source position it links to. The range of l_i is from 0 to K and there are M of these links. The source word position 0 is used to indicate NULL which we imagine gives rise to unaligned English words. In this paper, we refer to these words as being spontaneous. A valid link configuration has M links. Define $\mathcal{L}$ to be the set of all possible valid link configurations, and L to be a member of that set. We seek to maximize the alignment probability by finding the optimum link configuration L_{opt},

$$\begin{aligned} p(L_{\text{opt}}|S,T) &= \arg\max_{L\in\mathcal{L}} p(L|S,T) \\ &= p(l_i^M|t_1^M, s_1^K) \\ &= \prod_{i=0}^{M} p(l_i|t_1^M, s_1^K, l_1^{i-1}). \end{aligned}$$

We factor this into a transition model and an observation model,

$$p(L|S,T) = \frac{1}{Z}\prod_{i=0}^{M} p(l_i|l_{i-1})^{\alpha} p(l_i|t_1^M, s_1^K, l_1^{i-1})^{1-\alpha}.$$

where Z is the normalizing constant.

We factor the model as above so that the transition model computation, which uses information available on the search hypotheses, is reduced during the search process. In the aligner presented here, α is always set to 0.5. Next we will describe the transition model, then the observation model and finally the experiments in alignment and machine translation.

In the IBM Model 1 aligner, the choice of the language to serve as states of the search algorithm is not prescribed, but practically the choice is important as it affects performance. To see this, note that in generative models an input word can only be aligned to a single state in the search. In our current situation, we are interested in aligning unsegmented Arabic words and typical words have a few affixes to indicate for example pronouns, definiteness, prepositions and conjunctions. In English these are separate words, and therefore to maximize performance the unsegmented Arabic words serve as states in the search algorithm and we align English words to these states.

3.1 Transition Model

The transition model tends to keep the alignments close together and penalizes alignments in which adjacent words in the target language come from very distant words in the source language. Also, we would like to penalize many English words coming from the same Arabic state; we call this the state visit penalty and will be described later. In this paper, we use a parametric form for the transition model,

$$p(l_i|l_{i-1}) = \frac{1}{Z(l_{i-1})}\left[\frac{1}{\text{dist}(l_i, l_{i-1})} + \frac{1}{ns(l_i)}\right] \quad (1)$$

where $ns(i)$ represents the state visit penalty for state i, $Z(l_{i-1})$ is the normalization constant and

$$\text{dist}(l_i, l_{i-1}) = \min(|l_i - l_{i-1}|, |l_i - f_i|) + a.$$

Here a is a penalty for a zero distance transition and is set to 1 in the experiments below. The min operator chooses the lowest cost transition distance either from the previous state or the frontier state, f_i, which is the right most state that has been visited (even though Arabic is normally displayed right to left, we make our Arabic state graphs from left to right). This is a language specific criteria and intended to model the adjective noun reversal between English and Arabic. Once the current noun phrase is completed, the next word often aligns to the state just beyond frontier state. As an example, in Figure 1, the verb 'pointed' aligns to the first Arabic word 'wA$Art', and aligning the 'to' to its Arabic counterpart 'Aly' would incur normally a distance of 3 but with the frontier notion it incurs only a penalty of 1 on the hypothesis that aligns the word 'second' to 'AlvAnyp'. In this alignment with the frontier notion, there are only distance 1 transitions, whereas the traditional shapes would incur a penalty of 2 for alignment of 'pointed' and a penalty of 3 for the word 'to'.

The state visit penalty, $ns(i)$ is the distance between the English words aligned to this state times the number of state visits[1]. This penalty controls the fertility of the Arabic words. To determine the English words that aligned to the Arabic position, the search path is traced back for each hypothesis and a sufficiently large beam is maintained so that alignments in the future can correct past alignment decisions. This penalty allows English determiners and prepositions to align to the Arabic content word while penalizing distant words from aligning to the state. In terms of alignment F-measure to be described below, the state visit penalty, if removed makes the performance degrade from F=87.8 to F=84.0 compared to removing the frontier notion which only degrades performance to F=86.9.

3.2 Observation Model

The observation model measures the linkage of the source and target using a set of feature functions defined on the words and their context. In Figure 1, an event is a single link from an English word to an Arabic state and the event space is the sentence pair. We use the maximum entropy formulation (e.g. (Berger et al., 1996)),

$$f = \psi(l_i)$$
$$h = \left[t_1^{i-1}, s_1^K\right]$$
$$p(f|h) = \frac{1}{Z(h)} \exp \sum_i \lambda_i \phi_i(h, f),$$

where Z(h) is the normalizing constant,

$$Z(h) = \sum_f \exp \sum_i \lambda_i \phi_i(h, f).$$

and $\phi_i(h, f)$ are binary valued feature functions. The function ψ selects the Arabic word at the position being linked or in the case of segmentation features, one of the segmentations of that position. We restrict the history context to select from the current English word and words to the left as well as the current word's WordNet (Miller, 1990) synset as required by the features defined below. As in (Cherry and Lin, 2003), the above functions simplify the conditioning portion, h by utilizing only the words and context involved in the link l_i. Training is done using the IIS technique (Della Pietra et al., 1995) and convergence often occurs in 3-10 iterations. The five types of features which are utilized in the system are described below.

Phrase to phrase (for example, idiomatic phrases) alignments are intepreted as each English word coming from each of the Arabic words.

3.2.1 Lexical Features

The lexical features are similar to the translation matrix of the IBM Model 1. However, there is a signficant out of vocabulary (OOV) issue in the model since training data is limited. All words that have a corpus frequency of 1 are left out of the model and classed into an unknown word class in order to explicitly model connecting unknown words. From the training data we obtain 50K lexical features, and applying the Arabic segmenter obtain another 17K lexical features of the form ϕ(English content word, Arabic stem).

3.2.2 Arabic Segmentation Features

An Arabic segmenter similar to (Lee et al., 2003) provides the segmentation features. A small dictionary is used (with 71 rules) to restrict the set of Arabic segments that can align to English stopwords, for example that 'the' aligns to 'Al#' and that 'for', 'in' and 'to' align to 'b#' and 'her' aligns with the suffix '+hA'. Segmentation features also help align unknown words, as stems might be seen in the training corpus with other prefixes or suffixes. Additionally, the ability to align the prefix and suffix accurately, tends to 'drag' the unknown stem to its English target.

[1]We are overloading the word 'state' to mean Arabic word position.

3.2.3 WordNet Features

WordNet features provide normalization on the English words. The feature is instantiated for nouns, adjectives, adverbs and verbs following their definitions in WordNet. If the Arabic word has a segmentation then the feature is ϕ(WordNet synset id, Arabic stem), otherwise it is ϕ(WordNet synset id, Arabic word). The feature ties together English synonyms and helps improve recall of the aligner.

3.2.4 Spelling Feature

The spelling feature is applied only on unknown words and is used to measure the string kernel distance(Lodhi et al., 2000) between romanized Arabic and English words. The feature is designed primarily to link unknown names. For example, 'Clinton' is written as 'klyntwn' in one of its romanized Arabic versions. In a sentence, measuring the string kernel distance shows a correlation between these names even though there is not much overlap between the characters. The feature has four possible values: nomatch, somematch, goodmatch, and exact.

3.2.5 Dynamic Features

Dynamic features are defined on the lattice of the search algorithm. These features fire when the previous source and target word pair are linked. For example, one such feature is 'b# in' and if on the hypothesis we have just linked this pair and the next English word is being aligned to the stem of the Arabic word where this prefix occurs, this feature fires and boosts the probability that the next words are aligned. The basic intuition behind this feature is that words inside prepositional phrases tend to align, which is similar to the dependency structure feature of (Cherry and Lin, 2003).

At training time, the lattice reduces to the single path provided by the annotation. Since this feature tends to suffer from the drag of function words, we insist that the next words that are being linked have at least one feature that applies. All word pairs linked in the training data have lexical features as described above, and if both source and target words are unknown they have a single feature for their link. Applying dynamic features on words that have at least one other feature prevents words which are completely unrelated from being linked because of a feature about the context of the words.

Two types of dynamic features are distinguished: (a) English word with Arabic prefix/suffix and (b) English word with Arabic stem.

4 Smoothing the Observation Model

Since the annotated training data for word alignment is limited and a much larger parallel corpus is available for other aligners, we smooth the observation probability with an IBM Model 1 estimate,

$$p(l_i|t_1^M, s_1^K) = \frac{1}{Z} p_{\mathrm{ME}}(l_i|t_1^M, s_1^K)^{\beta} p_{\mathrm{M1}}(s|t_i)^{1-\beta}.$$

where β is set to 0.9 in the experiments below. In the equation above, the s represents the Arabic word that is being linked from the English word t_i.

When β is set to 1.0 there is no smoothing performed and performance degrades to F=84.0 from the best system performance (F=87.8). When β is set to 0, the model uses only the IBM Model 1 distribution and the resulting aligner is similar to an HMM aligner with the transition shape discussed above and yields performance of F=73.2.

	Anno. 1 Correction	Anno. 1'	Anno. 2
Anno. 1	96.5	92.4	91.7
Anno. 1'	95.2	—	93.2

Table 1: F-measure for human performance on word alignment for Arabic-English.

5 Search Algorithm

A beam search algorithm is utilized with the English words consumed in sequence and the Arabic word positions serving as states in the search process. In order to take advantage of the transition model described above, a large beam must be maintained. To see this, note that English words often repeat in a sentence and the models will tend to link the word to all Arabic positions which have the same Arabic content. In traditional algorithms, the Markov assumption is made and hypothesis are merged if they have the same history in the previous time step. However, here we maintain all hypotheses and merge only if the paths are same for 30 words which is the average sentence length.

6 Experimental Data

We have word aligned a portion of the Arabic Treebank (4300 sentences) and material from the LDC news sources (LDC, 2005) to obtain a total of 10.3K sentence pairs for training. As a test of alignment, we use the first 50 sentences of the MT03 Evaluation test set which has 1313 Arabic words and 1528 English words [2]. In terms of annotation guidelines, we use the following instructions: (a) Align determiners to their head nouns, (b) Alignments are done word by word unless the phrase is idiomatic in which case the entire phrase to phrase alignment was marked, (c) spontaneous words are marked as being part of a

[2]The test data is available by contacting the authors.

	1K	3K	5K	7K	9K	10.3K
# of features	15510	32111	47962	63140	73650	80321
English % OOV	15.9	8.2	5.5	4.4	4.05	3.6
Arabic % OOV	31	19.6	15.6	13.2	10.8	10.3
F-measure	83.2	85.4	86.5	87.4	87.5	87.8

Table 2: Varying Training data size.

phrase wherever possible but left unaligned if there is no evidence to link the word.

In order to measure alignment performance, we use the standard AER measure (Och and Ney, 2000) but consider all links as sure. This measure is then related to the F-measure which can be defined in terms of precision and recall as

Precision The number of correct word links over the total number of proposed links.

Recall The number of correct word links over the total number of links in the reference.

and the usual definition of the F-measure,

$$F = \frac{2PR}{(R+P)}$$

and define the alignment error as AER $= 1 - F$. In this paper, we report our results in terms of F-measure over aligned links. Note that links to the NULL state (unaligned English words) are not included in the F-measure. Systems are compared relative to the reduction in AER.

6.1 Annotator Agreement

We measure intra/inter-annotator agreement on the test set in order to determine the feasibility of human annotation of word links. These are shown in Table 1. In the table, the column for 'Annotator 1 Correction' is the first annotator correcting his own word alignments after a span of a year. After two weeks, the annotator (Annotator 1') was given the same material with all the links removed and asked to realign and we see that there is more discrepancy in resulting alignments. The differences are largely on the head concept where determiners are attached and the alignment of spontaneous words. The performance with a second annotator is in the same range as the reannotation by a single annotator.

7 Experiments

In order to evaluate the performance of the algorithm, we investigate the effect due to: (a) increasing the training data size, (b) additional feature types, and (c) comparable algorithms.

7.1 Training Data Size

We varied the training data size from 1K sentences to the complete set in Table 2. Each batch re-estimates the unknown word class by creating a vocabulary on the training set. The trend indicates a reasonable progression of performance and more data is required to determine the saturation point.

7.2 Feature Types

The results obtained by different feature sets are shown in Table 3. Each feature type was added incrementally (Add Feature column) to the line above to determine the effect of the individual feature types and then removed incrementally from the full system (Subtract Feature column) in order to see the final effect. The results indicate that lexical features are the most important type of feature; segmentation features further reduce the AER by 15.8%. The other features add small gains in performance which, although are not statistically significant for the alignment F-measure, are important in terms of feature extraction. Segmentation features discussed above result in both suffix and prefix features as well as stem features. In the Subtract column, for the segmentation feature, only the suffix and prefix features were removed. This result indicates that most of the alignment improvement from the segmentation feature comes in the form of new lexical features to link Arabic stems and English words.

7.3 Comparison to other alignment algorithms

In order to gauge the performance of the algorithm with respect to other alignment strategies, we provide results using GIZA++ and an HMM Max Posterior Algorithm (Ge, 2004). These algorithms, as well as the Model 1 smoothing for the MaxEnt aligner, are all trained on a corpus of 500K sentence pairs from the UN parallel corpus and the LDC news corpora released for 2005 (LDC, 2005). Note that these algorithms are unsupervised by design but we utilize them to have a baseline for comparing the performance of this supervised approach.

7.3.1 HMM Max Posterior Aligner

The maximum-posterior word alignments are obtained by finding the link configuration that maxi-

System	# of feats	Add Feature	Subtract Feature
Word pairs	50070	85.03	76.3
Spelling	4	85.11	87.7
Segmentation	70	87.39	87.5(*)
WordNet	13789	87.54	87.5
Dynamic-Words	1952	87.80	87.1
Dynamic-Segmentation	42	87.84	87.8

Table 3: Alignment performance in terms of the feature types utilized.

	F-Measure
GIZA++	79.5
HMM	76.3
MaxEnt	87.8

Table 4: Alignment performance

mizes the posterior state probability. In contrast, in performing a Viterbi alignment, we compute the best state sequence given the observation. The maximum posterior computes the best state one at a time and iterates over all possible combinations. Once we find the maximum in the posterior probability matrix, we also know the corresponding state and observation which is nothing but the word pair (s_j, t_i). We will then align the pair and continue to find the next posterior maximum and align the resulting pair. At each iteration of the process, a word pair is aligned. The process is repeated until either every word in one (or both) language is aligned or no more maximum can be found, whichever happens first.

7.3.2 GIZA Alignment

In order to contrast our algorithm, we ran GIZA++ in the standard configuration which implies 5 iterations of IBM Model 1, HMM, Model 3 and Model 4. All parameters are left to their default values.

The results using the three different aligners is shown in Table 4. The reduction in AER over the GIZA++ system is 40.5% and over the HMM system is 48.5%. The Wilcoxon signed-rank test yields a probability of 0.39 for rejecting the GIZA++ alignment over the HMM alignment, whereas the MaxEnt algorithm should be rejected with a probability of 1.7e-6 over the HMM algorithm and similarly MaxEnt should be rejected with a probability of 0.9e-6 over the GIZA++ algorithm. These significance tests indicate that the MaxEnt algorithm presented above is significantly better than either GIZA++ or HMM.

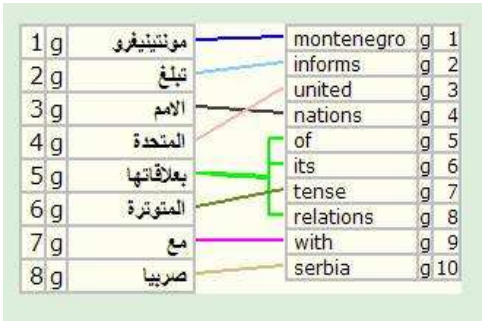

Figure 2: An alignment showing a split link from an Arabic word.

8 Phrase Extraction

Once an alignment is obtained, phrases which satisfy the inverse projection constraint are extracted (although earlier this constraint was called consistent alignments (Och et al., 1999)). This constraint enforces that a sequence of source words align to a sequence of target words as defined by the lowest and highest target index, and when the target words are projected back to the source language through the alignment, the original source sequence is retrieved. Examination of the hand alignment training data showed that this criteria is often violated for Arabic and English. Prepositional phrases with adjectives often require a split– for example, the alignment shown in Figure 2 has 'of its relations' aligned to a word in Arabic and 'tense' aligned to the next word. The inverse projection constraint fails in this case, and in the experiments below, we relax this constraint and generate features for single source words as long as the target phrase has a gap less than 2 English words. This relaxation allows a pair of adjectives to modify the head noun. In future work we explore the use of features with variables to be filled at decode time.

9 Translation Experiments

The experiments in machine translation are carried out on a phrase based decoder similar to the one de-

	MT03	MT04	MT05
GIZA++	0.454	—	—
HMM	0.459	0.419	0.456
MaxEnt	0.468	0.433	0.451
Combined	0.479	0.437	0.465
Significance	0.017	0.020	—

Table 5: Machine Translation Performance using the NIST 2005 Bleu scorer

scribed in (Tillmann and Ney, 2003). In order to contrast the performance of the extracted features, we compare the translation performance to (a) a system built from alignments proposed by an HMM Max Posterior Aligner, and (b) a system built from GIZA alignments. All other parameters of the decoder remain constant and only the feature set is changed for these experiments. As training data, we use the UN parallel corpus and the LDC news corpora released in 2005. Comparison should therefore be only made across systems reported here and not to earlier evaluations or other systems. The results are shown in Table 5.

Combination of the phrasal features from the HMM and MaxEnt alignments results in the 'Combined' system. The Combined system performs better in all cases; in MT03 and MT04 the MaxEnt derived features perform better than the HMM system. In MT05, there is a slight degradation which is not significant and the combination system still results in an improvement over either system. Since the MaxEnt aligner has access to a unique resource, every attempt was made to make that resource available to the other systems. Although GIZA++ and HMM can not directly utilize word aligned data, the training data for MaxEnt was converted to parallel sentences where each sentence has only the pair of linked words. The resulting numbers make both HMM and GIZA much closer in performance to the MaxEnt aligner but the results are better for comparing alignment methods.

10 Error Analysis and Discussion

The alignment errors made by the system can be attributed to

- English words that require multi-word Arabic states, for example (a) dates which are written in Arabic in more than one form 'kAnwn Alvany / ynAyr' for 'january', and (b) compound words like 'rAm Allh' in English is 'Ramallah'.

- Rare translation of a common Arabic word as well as a common English word used as the translation for a rare Arabic word.

- Parallel corpora mismatch: training material for translation is processed at a document level and yet systems often operate at a sentence level. Human translators often use pronouns for earlier mentioned names although in the source language the name is repeated. Information which is sometimes repeated in the source in an earlier sentence is dropped in future sentences of the document. Document level features are required to allow the system to have information to leave these words unaligned.

Figure 3 shows a human alignment on the left and a machine output on the right. The columns next to the words indicate whether the alignments are 'good' or 'extra' which indicates that these words are aligned to the special NULL state. There are two examples of multi-word Arabic states shown: (a) for 'january', and (b) the English word 'agenda'. The system aligns 'the' before committee and it seems in this case its an annotation error. In this example the Arabic words lnAHyp, AltnZym, wAlAEdAd and Allwjsty are all unknown words in the vocabulary yet the system managed to link 3 out 4 words correctly.

While significant gains have been made in alignment performance, these gains have not directly translated to machine translation improvements. In fact, although the GIZA system is better than the HMM system at alignment, the machine translation result on MT03 indicates a slight degradation (although it is not statistically significant). The prime reason for this is that features extracted from the alignments are aggregated over the training corpus and this process helps good alignments to have significantly better counts than errors in alignment. Aligning rare words correctly should help performance but since their count is low it is not reflected in bleu scores.

11 Conclusion and Future Work

This paper presented a word aligner trained on annotated data. While the performance of the aligner is shown to be significantly better than other unsupervised algorithms, the utility of these alignments in machine translation is still an open subject although gains are shown in two of the test sets. Since features are extracted from a parallel corpus, most of the information relating to the specific sentence alignment is lost in the aggregation of features across sentences. Improvements in capturing sentence context could allow the machine translation system to use a rare but correct link appropriately.

Another significant result is that a small amount (5K sentences) of word-aligned data is sufficient for this algorithm since a provision is made to handle

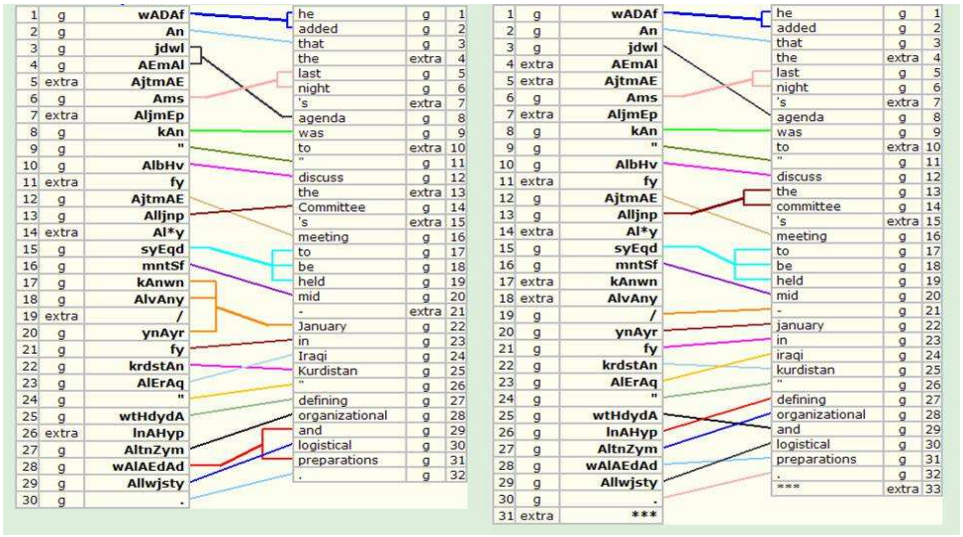

Figure 3: An example sentence with human output on the left and system output on the right.

unknown words appropriately.

12 Acknowledgements

This work was partially supported by the Defense Advanced Research Projects Agency and monitored by SPAWAR under contract No. N66001-99-2-8916. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the U.S. government and no official endorsement should be inferred. This paper owes much to the collaboration of the Statistical MT group at IBM.

References

Adam L. Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

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. 2003. A probability model to improve word alignment. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Sapporo, Japan.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1995. Inducing features of random fields. *Technical Report, Department of Computer Science, Carnegie-Mellon University, CMU-CS-95-144*, May.

Niyu Ge. 2004. Improvement in Word Alignments. *Presentation given at DARPA/TIDES MT workshop.*

LDC. 2005. http://ldc.upenn.edu/projects/tides/mt2005ar.htm.

Young-Suk Lee, Kishore Papineni, and Salim Roukos. 2003. Language model based arabic word segmentation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 399–406, Sapporo, Japan.

Huma Lodhi, John Shawe-Taylor, Nello Cristianini, and Christopher J. C. H. Watkins. 2000. Text classification using string kernels. In *NIPS*, pages 563–569.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Robert C. Moore. 2004. Improving IBM Word-Alignment Model 1. In *42nd Annual Meeting of the Association for Computational Linguistics*, pages 518–525, Barcelona, Spain.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China.

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

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, College Park, Maryland.

Franz Josef Och. 2003. Minimum error rate training in Statistical Machine Translation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for Statistical Machine Translation. 29(1):97–133.

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM BasedWord Alignment in Statistical Machine Translation. In *Proc. of the 16th Int. Conf. on Computational Linguistics (COLING 1996)*, pages 836–841, Copenhagen, Denmark, August.