

A New Objective Function for Word Alignment

Tugba Bodrumlu Kevin Knight Sujith Ravi
Information Sciences Institute & Computer Science Department
University of Southern California
bodrumlu@usc.edu, knight@isi.edu, sravi@isi.edu

Abstract

We develop a new objective function for word alignment that measures the size of the bilingual dictionary induced by an alignment. A word alignment that results in a small dictionary is preferred over one that results in a large dictionary. In order to search for the alignment that minimizes this objective, we cast the problem as an integer linear program. We then extend our objective function to align corpora at the sub-word level, which we demonstrate on a small Turkish-English corpus.

1 Introduction

Word alignment is the problem of annotating a bilingual text with links connecting words that have the same meanings. Figure 1 shows sample input for a word aligner (Knight, 1997). After analyzing the text, we may conclude, for example, that *sprok* corresponds to *dat* in the first sentence pair.

Word alignment has several downstream consumers. One is machine translation, where programs extract translation rules from word-aligned corpora (Och and Ney, 2004; Galley et al., 2004; Chiang, 2007; Quirk et al., 2005). Other downstream processes exploit dictionaries derived by alignment, in order to translate queries in cross-lingual IR (Schönhofen et al., 2008) or re-score candidate translation outputs (Och et al., 2004).

Many methods of automatic alignment have been proposed. Probabilistic generative models like IBM 1-5 (Brown et al., 1993), HMM (Vogel et al., 1996), ITG (Wu, 1997), and LEAF (Fraser and Marcu, 2007) define formulas for $P(f | e)$ or $P(e, f)$, with

ok-voon ororok sprok
at-voon bichat dat

erok sprok izok hihok ghirok
totat dat arrat vat hilat

ok-drubel ok-voon anak plok sprok
at-drubel at-voon pippat rrat dat

ok-voon anak drok brok jok
at-voon krat pippat sat lat

wiwok farok izok stok
totat jjat quat cat

lalok sprok izok jok stok
wat dat krat quat cat

lalok farok ororok lalok sprok izok enemok
wat jjat bichat wat dat vat eneat

lalok brok anak plok nok
iat lat pippat rrat nnat

wiwok nok izok kantok ok-yurp
totat nnat quat oloat at-yurp

lalok mok nok yorok ghirok klok
wat nnat gat mat bat hilat

lalok nok crrrok hihok yorok zanzanak
wat nnat arrat mat zanzanat

lalok rarok nok izok hihok mok
wat nnat forat arrat vat gat

Figure 1: Word alignment exercise (Knight, 1997).

hidden alignment variables. EM algorithms estimate dictionary and other probabilities in order to maximize those quantities. One can then ask for Viterbi alignments that maximize $P(\text{alignment} \mid e, f)$. Discriminative models, e.g. (Taskar et al., 2005), instead set parameters to maximize alignment accuracy against a hand-aligned development set. EMD training (Fraser and Marcu, 2006) combines generative and discriminative elements.

Low accuracy is a weakness for all systems. Most practitioners still use 1990s algorithms to align their data. It stands to reason that we have not yet seen the last word in alignment models.

In this paper, we develop a new objective function for alignment, inspired by watching people manually solve the alignment exercise of Figure 1. When people attack this problem, we find that once they create a bilingual dictionary entry, they like to *re-use* that entry as much as possible. Previous machine aligners emulate this to some degree, but they are not explicitly programmed to do so.

We also address another weakness of current aligners: they only align full words. With few exceptions, e.g. (Zhang et al., 2003; Snyder and Barzilay, 2008), aligners do not operate at the sub-word level, making them much less useful for agglutinative languages such as Turkish.

Our present contributions are as follows:

- We offer a simple new objective function that scores a corpus alignment based on how many distinct bilingual word pairs it contains.
- We use an integer programming solver to carry out optimization and corpus alignment.
- We extend the system to perform sub-word alignment, which we demonstrate on a Turkish-English corpus.

The results in this paper constitute a proof of concept of these ideas, executed on small corpora. We conclude by listing future directions.

2 New Objective Function for Alignment

We search for the legal alignment that *minimizes the size of the induced bilingual dictionary*. By dictionary size, we mean the number of distinct word-pairs linked in the corpus alignment. We can immediately investigate how different alignments stack up, according to this objective function. Figure 2

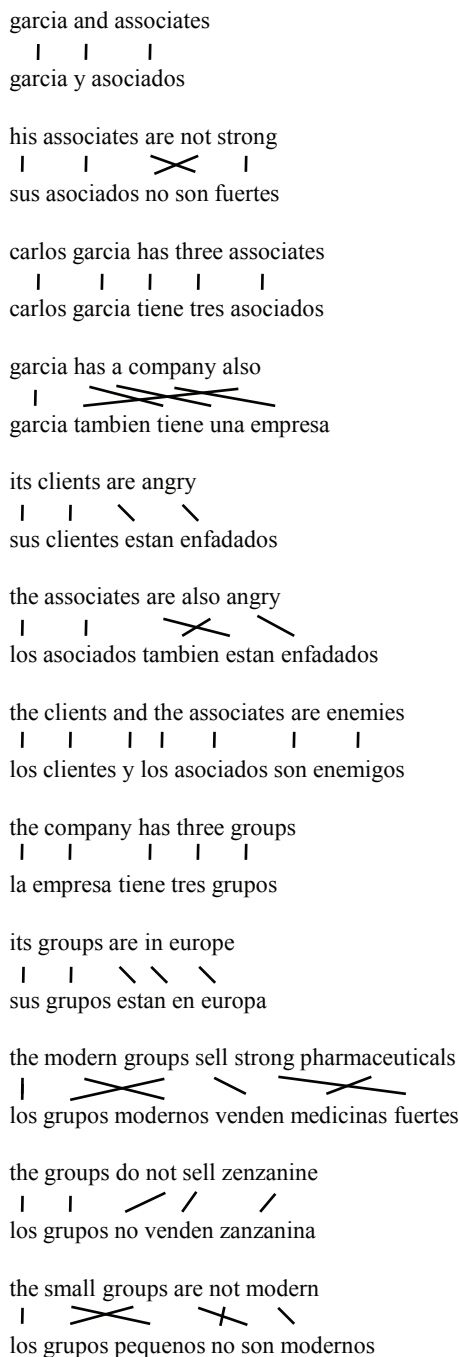


Figure 2: Gold alignment. The induced bilingual dictionary has 28 distinct entries, including *garcia/garcia*, *are/son*, *are/estan*, *not/no*, *has/tiene*, etc.

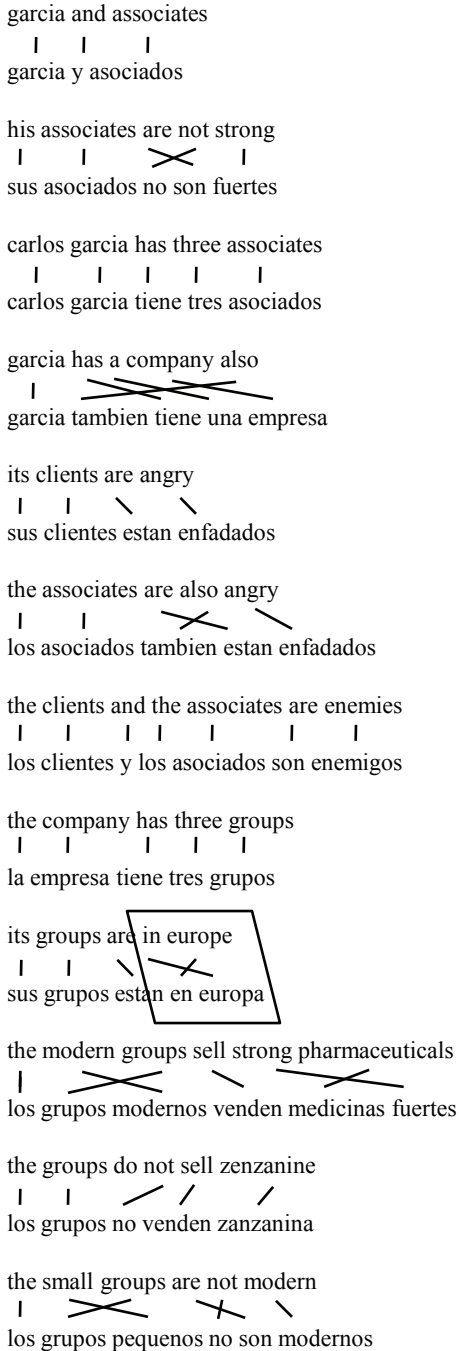


Figure 3: IP alignment. The induced bilingual dictionary has 28 distinct entries.

shows the gold alignment for the corpus in Figure 1 (displayed here as English-Spanish), which results in 28 distinct bilingual dictionary entries. By contrast, a monotone alignment induces 39 distinct entries, due to less re-use.

Next we look at how to automatically rifle through all legal alignments to find the one with the best score. What is a legal alignment? For now, we consider it to be one where:

- Every foreign word is aligned exactly once (Brown et al., 1993).
- Every English word has either 0 or 1 alignments (Melamed, 1997).

We formulate our integer program (IP) as follows. We set up two types of binary variables:

- Alignment link variables. If $link-i-j-k = 1$, that means in sentence pair i , the foreign word at position j aligns to the English words at position k .
- Bilingual dictionary variables. If $dict-f-e = 1$, that means word pair (f, e) is “in” the dictionary.

We constrain the values of *link* variables to satisfy the two alignment conditions listed earlier. We also require that if $link-i-j-k = 1$ (i.e., we’ve decided on an alignment link), then $dict-f_{ij}-e_{ik}$ should also equal 1 (the linked words are recorded as a dictionary entry).¹ We do not require the converse—just because a word pair is available in the dictionary, the aligner does not have to link every instance of that word pair. For example, if an English sentence has two *the* tokens, and its Spanish translation has two *la* tokens, we should not require that all four links be active—in fact, this would conflict with the 1-1 *link* constraints and render the integer program unsolvable. The IP reads as follows:

$$\begin{aligned}
 &\text{minimize:} \\
 &\quad \sum_{f,e} dict-f-e \\
 &\text{subject to:} \\
 &\quad \forall_{i,j} \sum_k link-i-j-k = 1 \\
 &\quad \forall_{i,k} \sum_j link-i-j-k \leq 1 \\
 &\quad \forall_{i,j,k} link-i-j-k \leq dict-f_{ij}-e_{ik}
 \end{aligned}$$

On our Spanish-English corpus, the *cplex*² solver obtains a minimal objective function value of 28. To

¹ f_{ij} is the j th foreign word in the i th sentence pair.

² www.ilog.com/products/cplex

get the second-best alignment, we add a constraint to our IP requiring the sum of the n variables active in the previous solution to be less than n , and we re-run *cplex*. This forces *cplex* to choose different variable settings on the second go-round. We repeat this procedure to get an ordered list of alignments.³

We find that there are 8 distinct solutions that yield the same objective function value of 28. Figure 3 shows one of these. This alignment is not bad, considering that word-order information is not encoded in the IP. We can now compare several alignments in terms of both dictionary size and alignment accuracy. For accuracy, we represent each alignment as a set of tuples $\langle i, j, k \rangle$, where i is the sentence pair, j is a foreign index, and k is an English index. We use these tuples to calculate a balanced f-score against the gold alignment tuples.⁴

Method	Dict size	f-score
Gold	28	100.0
Monotone	39	68.9
IBM-1 (Brown et al., 1993)	30	80.3
IBM-4 (Brown et al., 1993)	29	86.9
IP	28	95.9

The last line shows an average f-score over the 8 tied IP solutions.

Figure 4 further investigates the connection between our objective function and alignment accuracy. We sample up to 10 alignments at each of several objective function values v , by first adding a constraint that *dict* variables add to exactly v , then iterating the n-best list procedure above. We stop when we have 10 solutions, or when *cplex* fails to find another solution with value v . In this figure, we see a clear relationship between the objective function and alignment accuracy—minimizing the former is a good way to maximize the latter.

³This method not only suppresses the IP solutions generated so far, but it suppresses additional solutions as well. In particular, it suppresses solutions in which all *link* and *dict* variables have the same values as in some previous solution, but some additional *dict* variables are flipped to 1. We consider this a feature rather than a bug, as it ensures that all alignments in the n-best list are unique. For what we report in this paper, we only create n-best lists whose elements possess the same objective function value, so the issue does not arise.

⁴ P = proportion of proposed links that are in gold, R = proportion of gold links that are proposed, and f-score = $2PR/(P+R)$.

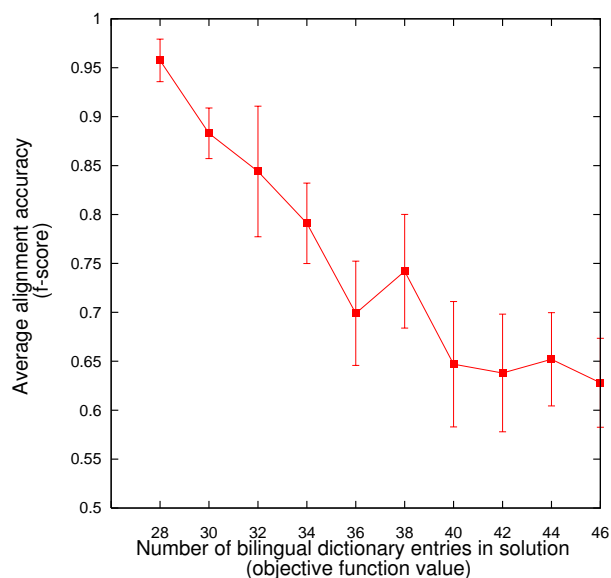


Figure 4: Relationship between IP objective (x-axis = size of induced bilingual dictionary) and alignment accuracy (y-axis = f-score).

Turkish	English
yururum	i walk
yururler	they walk

Figure 5: Two Turkish-English sentence pairs.

3 Sub-Word Alignment

We now turn to alignment at the sub-word level. Agglutinative languages like Turkish present challenges for many standard NLP techniques. An agglutinative language can express in a single word (e.g., *yurumuyoruz*) what might require many words in another language (e.g., *we are not walking*). Naively breaking on whitespace results in a very large vocabulary for Turkish, and it ignores the multi-morpheme structure inside Turkish words.

Consider the tiny Turkish-English corpus in Figure 5. Even a non-Turkish speaker might plausibly align *yurur* to *walk*, *um* to *I*, and *ler* to *they*. However, none of the popular machine aligners is able to do this, since they align at the whole-word level. Designers of translation systems sometimes employ language-specific word breakers before alignment, though these are hard to build and maintain, and they are usually not only language-specific, but also language-pair-specific. Good un-

supervised monolingual morpheme segmenters are also available (Goldsmith, 2001; Creutz and Lagus, 2005), though again, these do not do joint inference of alignment and word segmentation.

We extend our objective function straightforwardly to sub-word alignment. To test our extension, we construct a Turkish-English corpus of 1616 sentence pairs. We first manually construct a regular tree grammar (RTG) (Gecseg and Steinby, 1984) for a fragment of English. This grammar produces English trees; it has 86 rules, 26 states, and 53 terminals (English words). We then construct a tree-to-string transducer (Rounds, 1970) that converts English trees into Turkish *character* strings, including space. Because it does not explicitly enumerate the Turkish vocabulary, this transducer can output a very large number of distinct Turkish words (i.e., character sequences preceded and followed by space). This transducer has 177 rules, 18 states, and 23 terminals (Turkish characters). RTG generation produces English trees that the transducer converts to Turkish, both via the tree automata toolkit Tiburon (May and Knight, 2006). From this, we obtain a parallel Turkish-English corpus. A fragment of the corpus is shown in Figure 6. Because we will concentrate on finding Turkish sub-words, we manually break off the English sub-word *-ing*, by rule, as seen in the last line of the figure.

This is a small corpus, but good for demonstrating our concept. By automatically tracing the internal operation of the tree transducer, we also produce a gold alignment for the corpus. We use the gold alignment to tabulate the number of morphemes per Turkish word:

n	% Turkish types with n morphemes	% Turkish tokens with n morphemes
1	23.1%	35.5%
2	63.5%	61.6%
3	13.4%	2.9%

Naturally, these statistics imply that standard whole-word aligners will fail. By inspecting the corpus, we find that 26.8 is the maximum f-score available to whole-word alignment methods.

Now we adjust our IP formulation. We broaden the definition of legal alignment to include breaking any foreign word (token) into one or more sub-word (tokens). Each resulting sub-word token is aligned

to exactly one English word token, and every English word aligns to 0 or 1 foreign sub-words. Our *dict-f-e* variables now relate Turkish sub-words to English words. The first sentence pair in Figure 5 would have previously contributed two *dict* variables; now it contributes 44, including things like *dict-uru-walk*. We consider an alignment to be a set of tuples $\langle i, j1, j2, k \rangle$, where $j1$ and $j2$ are start and end indices into the foreign character string. We create *align-i-j1-j2-k* variables that connect Turkish character spans with English word indices. Alignment variables constrain dictionary variables as before, i.e., an alignment link can only “turn on” when licensed by the dictionary.

We previously constrained every Turkish word to align to something. However, we do not want every Turkish character *span* to align—only the spans explicitly chosen in our word segmentation. So we introduce *span-i-j1-j2* variables to indicate segmentation decisions. Only when $span-i-j1-j2 = 1$ do we require $\sum_k align-i-j1-j2-k = 1$.

For a coherent segmentation, the set of active *span* variables must cover all Turkish letter tokens in the corpus, and no pair of spans may overlap each other. To implement these constraints, we create a lattice where each node represents a Turkish index, and each transition corresponds to a *span* variable. In a coherent segmentation, the sum of all *span* variables entering an lattice-internal node equals the sum of all *span* variables leaving that node. If the sum of all variables leaving the start node equals 1, then we are guaranteed a left-to-right path through the lattice, i.e., a coherent choice of 0 and 1 values for *span* variables.

The IP reads as follows:

minimize:

$$\sum_{f,e} dict-f-e$$

subject to:

$$\forall_{i,j1,j2} \sum_k align-i-j1-j2-k = span-i-j1-j2$$

$$\forall_{i,k} \sum_{j1,j2} align-i-j1-j2-k \leq 1$$

$$\forall_{i,j1,j2,k} align-i-j1-j2-k \leq dict-f_{i,j1,j2-e_{i,k}}$$

$$\forall_{i,j} \sum_{j3} span-i-j3-j = \sum_{j3} span-i-j-j3$$

$$\forall_{i,w} \sum_{j>w} span-i-w-j = 1$$

(w ranges over Turkish word start indices)

With our simple objective function, we obtain an f-score of 61.4 against the gold standard. Sample gold and IP alignments are shown in Figure 7.

Turkish	English
onlari gordum	i saw them
gidecekler	they will go
onu stadyumda gordum	i saw him in the stadium
ogretmenlerim tiyatroya yurudu	my teachers walked to the theatre
cocuklar yurudu	the girls walked
babam restorana gidiyor	my father is walk ing to the restaurant
...	...

Figure 6: A Turkish-English corpus produced by an English grammar pipelined with an English-to-Turkish tree-to-string transducer.

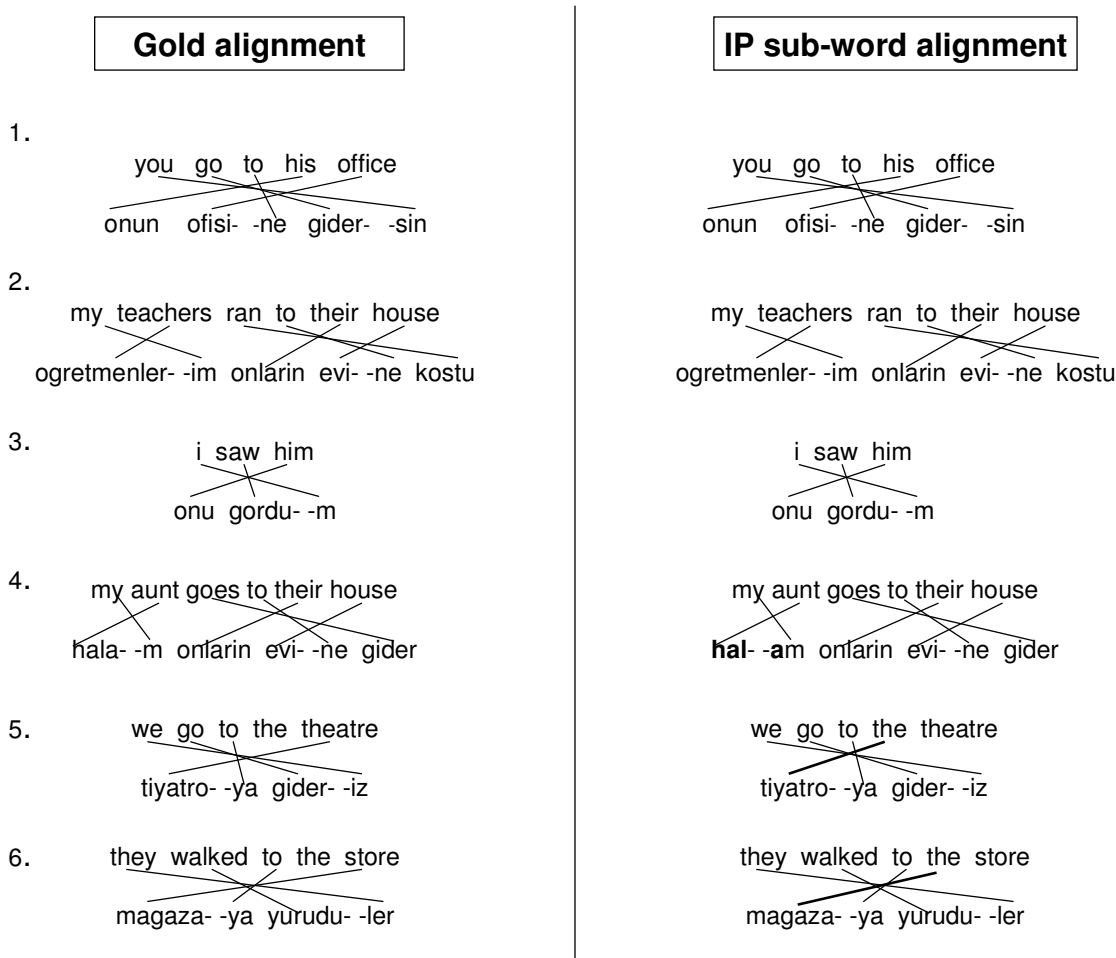


Figure 7: Sample gold and (initial) IP sub-word alignments on our Turkish-English corpus. Dashes indicate where the IP search has decided to break Turkish words in the process of aligning. For examples, the word *magazaya* has been broken into *magaza-* and *-ya*.

The last two incorrect alignments in the figure are instructive. The system has decided to align English *the* to the Turkish noun morphemes *tiyatrosu* and *magaza*, and to leave English nouns *theatre* and *store* unaligned. This is a tie-break decision. It is equally good for the objective function to leave *the* unaligned instead—either way, there are two relevant dictionary entries.

We fix this problem by introducing a special NULL Turkish token, and by modifying the IP to require every English token to align (either to NULL or something else). This introduces a cost for failing to align an English token x to Turkish, because a new $x/NULL$ dictionary entry will have to be created. (The NULL token itself is unconstrained in how many tokens it may align to.)

Under this scheme, the last two incorrect alignments in Figure 7 induce four relevant dictionary entries (*the/tiyatrosu*, *the/magaza*, *theatre/NULL*, *store/NULL*) while the gold alignment induces only three (*the/NULL*, *theatre/tiyatrosu*, *store/magaza*), because *the/NULL* is re-used. The gold alignment is therefore now preferred by the IP optimizer. There is a rippling effect, causing the system to correct many other decisions as well. This revision raises the alignment f-score from 61.4 to 83.4.

The following table summarizes our alignment results. In the table, “Dict” refers to the size of the induced dictionary, and “Sub-words” refers to the number of induced Turkish sub-word tokens.

Method	Dict	Sub-words	f-score
Gold (sub-word)	67	8102	100.0
Monotone (word)	512	4851	5.5
IBM-1 (word)	220	4851	21.6
IBM-4 (word)	230	4851	20.3
IP (word)	107	4851	20.1
IP (sub-word, initial)	60	7418	61.4
IP (sub-word, revised)	65	8105	83.4

Our search for an optimal IP solution is not fast. It takes 1-5 hours to perform sub-word alignment on the Turkish-English corpus. Of course, if we wanted to obtain optimal alignments under IBM Model 4, that would also be expensive, in fact NP-complete (Raghavendra and Maji, 2006). Practical Model 4

systems therefore make substantial search approximations (Brown et al., 1993).

4 Related Work

(Zhang et al., 2003) and (Wu, 1997) tackle the problem of segmenting Chinese while aligning it to English. (Snyder and Barzilay, 2008) use multilingual data to compute segmentations of Arabic, Hebrew, Aramaic, and English. Their method uses IBM models to bootstrap alignments, and they measure the resulting segmentation accuracy.

(Taskar et al., 2005) cast their alignment model as a minimum cost quadratic flow problem, for which optimal alignments can be computed with off-the-shelf optimizers. Alignment in the modified model of (Lacoste-Julien et al., 2006) can be mapped to a quadratic assignment problem and solved with linear programming tools. In that work, linear programming is not only used for alignment, but also for training weights for the discriminative model. These weights are trained on a manually-aligned subset of the parallel data. One important “mega” feature for the discriminative model is the score assigned by an IBM model, which must be separately trained on the full parallel data. Our work differs in two ways: (1) our training is unsupervised, requiring no manually aligned data, and (2) we do not bootstrap off IBM models. (DeNero and Klein, 2008) gives an integer linear programming formulation of another alignment model based on phrases. There, integer programming is used only for alignment, not for learning parameter values.

5 Conclusions and Future Work

We have presented a novel objective function for alignment, and we have applied it to whole-word and sub-word alignment problems. Preliminary results look good, especially given that new objective function is simpler than those previously proposed. The integer programming framework makes the model easy to implement, and its optimal behavior frees us from worrying about search errors.

We believe there are good future possibilities for this work:

- **Extend legal alignments to cover n-to-m and discontinuous cases.** While morpheme-to-morpheme alignment is more frequently a

1-to-1 affair than word-to-word alignment is, the 1-to-1 assumption is not justified in either case.

- **Develop new components for the IP objective.** Our current objective function makes no reference to word order, so if the same word appears twice in a sentence, a tie-break ensues.
- **Establish complexity bounds for optimizing dictionary size.** We conjecture that optimal alignment according to our model is NP-complete in the size of the corpus.
- **Develop a fast, approximate alignment algorithm for our model.**
- **Test on large-scale bilingual corpora.**

Acknowledgments

This work was partially supported under DARPA GALE, Contract No. HR0011-06-C-0022.

References

- P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2).
- D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).
- M. Creutz and K. Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proc. AKRR*.
- J. DeNero and D. Klein. 2008. The complexity of phrase alignment problems. In *Proc. ACL*.
- A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proc. ACL-COLING*.
- A. Fraser and D. Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proc. EMNLP-CoNLL*.
- M. Galley, M. Hopkins, K. Knight, and D. Marcu. 2004. What's in a translation rule. In *Proc. NAACL-HLT*.
- F. Gecseg and M. Steinby. 1984. *Tree automata*. Akademiai Kiado.
- J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2).
- K. Knight. 1997. Automating knowledge acquisition for machine translation. *AI Magazine*, 18(4).
- S. Lacoste-Julien, B. Taskar, D. Klein, and M. Jordan. 2006. Word alignment via quadratic assignment. In *Proc. HLT-NAACL*.
- J. May and K. Knight. 2006. Tiburon: A weighted tree automata toolkit. In *Proc. CIAA*.
- I. D. Melamed. 1997. A word-to-word model of translational equivalence. In *Proc. ACL*.
- F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).
- F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proc. HLT-NAACL*.
- C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. ACL*.
- U. Raghavendra and H. K. Maji. 2006. Computational complexity of statistical machine translation. In *Proc. EACL*.
- W. Rounds. 1970. Mappings and grammars on trees. *Theory of Computing Systems*, 4(3).
- P. Schönhofen, A. Benczúr, I. Bíró, and K. Csalogány. 2008. *Cross-language retrieval with wikipedia*. Springer.
- B. Snyder and R. Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL*.
- B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Proc. EMNLP*.
- S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proc. ACL*.
- D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3).
- Y. Zhang, S. Vogel, and A. Waibel. 2003. Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proc. Intl. Conf. on NLP and KE*.