Word Alignment via Submodular Maximization over Matroids

Hui Lin
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
hlin@ee.washington.edu

Jeff Bilmes
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
bilmes@ee.washington.edu

Abstract

We cast the word alignment problem as maximizing a submodular function under matroid constraints. Our framework is able to express complex interactions between alignment components while remaining computationally efficient, thanks to the power and generality of submodular functions. We show that submodularity naturally arises when modeling word fertility. Experiments on the English-French Hansards alignment task show that our approach achieves lower alignment error rates compared to conventional matching based approaches.

1 Introduction

Word alignment is a key component in most statistical machine translation systems. While classical approaches for word alignment are based on generative models (e.g., IBM models (Brown et al., 1993) and HMM (Vogel et al., 1996)), word alignment can also be viewed as a matching problem, where each word pair is associated with a score reflecting the desirability of aligning that pair, and the alignment is then the highest scored matching under some constraints.

Several matching-based approaches have been proposed in the past. Melamed (2000) introduces the competitive linking algorithm which greedily constructs matchings under the one-to-one mapping assumption. In (Matusov et al., 2004), matchings are found using an algorithm for constructing a maximum weighted bipartite graph matching (Schrijver, 2003), where word pair scores come from alignment posteriors of generative models. Similarly, Taskar et al. (2005) cast word alignment as a maximum weighted matching problem and propose a framework for learning word pair scores as a function of arbitrary features of that pair. These approaches, however, have two potentially substantial limitations: words have fertility of at most one, and interactions between alignment decisions are not representable.

Lacoste-Julien et al. (2006) address this issue by formulating the alignment problem as a quadratic assignment problem, and off-the-shelf integer linear programming (ILP) solvers are used to solve to optimization problem. While efficient for some median scale problems, ILP-based approaches are limited since when modeling more sophisticated interactions, the number of variables (and/or constraints) required grows polynomially, or even exponentially, making the resultant optimization impractical to solve.

In this paper, we treat the word alignment problem as maximizing a submodular function subject to matroid constraints (to be defined in Section 2). Submodular objective functions can represent complex interactions among alignment decisions, and essentially extend the modular (linear) objectives used in the aforementioned approaches. While our extensions add expressive power, they do *not* result in a heavy computational burden. This is because maximizing a monotone submodular function under a matroid constraint can be solved efficiently using a simple greedy algorithm. The greedy algorithm, moreover, is a constant factor approximation algorithm that guarantees a near-optimal solution. In this paper, we moreover show that submodularity naturally arises in word alignment problems when modeling word fertility (see Section 4). Experiment results on the English-French Hansards alignment task show that our approach achieves lower alignment error rates compared to the maximum weighted matching approach, while being at least 50 times

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 170–175,
Portland, Oregon, June 19-24, 2011.

faster than an ILP-based approach.

2 Background

Matroids and submodularity both play important roles in combinatorial optimization. We briefly introduce them here, referring the reader to (Schrijver, 2003) for details.

Matroids are combinatorial structures that generalize the notion of linear independence in matrices. A pair $(V, \mathcal{I})$ is called a *matroid* if V is a finite ground set and $\mathcal{I}$ is a nonempty collection of subsets of V that are *independent*. In particular, $\mathcal{I}$ must satisfy (i) if $X \subset Y$ and $Y \in \mathcal{I}$ then $X \in \mathcal{I}$, (ii) if $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then $X \cup \{e\} \in \mathcal{I}$ for some $e \in Y \setminus X$. We typically refer to a matroid by listing its ground set and its family of independent sets: $\mathcal{M} = (V, \mathcal{I})$.

A set function $f : 2^V \rightarrow \mathbb{R}$ is called *submodular* (Edmonds, 1970) if it satisfies the property of *diminishing returns*: for any $X \subseteq Y \subseteq V \setminus v$, a submodular function f must satisfy $f(X+v) - f(X) \geq f(Y+v) - f(Y)$. That is, the incremental "value" of v decreases as the context in which v is considered grows from X to Y. If this is satisfied everywhere with equality, then the function f is called *modular*. A set function f is *monotone nondecreasing* if $\forall X \subseteq Y$, $f(X) \leq f(Y)$. As shorthand, in this paper, monotone nondecreasing submodular functions will simply be referred to as *monotone submodular*.

Historically, submodular functions have their roots in economics, game theory, combinatorial optimization, and operations research. More recently, submodular functions have started receiving attention in the machine learning and computer vision community (Kempe et al., 2003; Narasimhan and Bilmes, 2004; Narasimhan and Bilmes, 2005; Krause and Guestrin, 2005; Narasimhan and Bilmes, 2007; Krause et al., 2008; Kolmogorov and Zabin, 2004; Jegelka and Bilmes, 2011) and have recently been introduced to natural language processing for the task of document summarization (Lin and Bilmes, 2010; Lin and Bilmes, 2011).

3 Approach

We are given a source language (English) string $e_1^I = e_1, \cdots, e_i, \cdots, e_I$ and a target language (French) string $f_1^J = f_1, \cdots, f_j, \cdots, f_J$ that have to be aligned. Define the word positions in the English string as set $E \triangleq \{1, \cdots, I\}$ and positions in the French string as set $F \triangleq \{1, \cdots, J\}$. An alignment A between the two word strings can then be seen as a subset of the Cartesian product of the word positions, i.e., $A \subseteq \{(i, j) : i \in E, j \in F\} \triangleq V$, and $V = E \times F$ is the ground set. For convenience, we refer to element $(i, j) \in A$ as an *edge* that connects i and j in alignment A.

Restricting the fertility of word f_j to be at most k_j is mathematically equivalent to having $|A \cap P_j^E| \leq k_j$, where $A \subseteq V$ is an alignment and $P_j^E = E \times \{j\}$. Intuitively, P_j^E is the set of all possible edges in the ground set that connect to j, and the cardinality of the intersection between A and P_j^E indicates how many edges in A are connected to j. Similarly, we can impose constraints on the fertility of English words by constraining the alignment A to satisfy $|A \cap P_i^F| \leq k_i$ for $i \in E$ where $P_i^F = \{i\} \times F$. Note that either of $\{P_j^E : j \in F\}$ or $\{P_i^F : i \in E\}$ constitute a partition of V. Therefore, alignments A that satisfy $|A \cap P_j^E| \leq k_j, \forall j \in F$, are independent in the *partition matroid* $\mathcal{M}_E = (V, \mathcal{I}_E)$ with

$$\mathcal{I}_E = \{A \subseteq V : \forall j \in F, |A \cap P_j^E| \leq k_j\},$$

and alignments A that satisfy $|A \cap P_i^F| \leq k_i, \forall i \in E$, are independent in matroid $\mathcal{M}_F = (V, \mathcal{I}_F)$ with

$$\mathcal{I}_F = \{A \subseteq V : \forall i \in E, |A \cap P_i^F| \leq k_i\}.$$

Suppose we have a set function $f : 2^V \rightarrow \mathbb{R}_+$ that measures quality (or scores) of an alignment $A \subseteq V$, then when also considering fertility constraints, we can treat the word alignment problem as maximizing a set function subject to matroid constraint:

Problem 1. $\max_{A \subseteq V} f(A)$, *subject to:* $A \in \mathcal{I}$,

where $\mathcal{I}$ is the set of independent sets of a matroid (or it might be the set of independent sets simultaneously in two matroids, as we shall see later).

Independence in partition matroids generalizes the typical matching constraints for word alignment, where each word aligns to at most one word ($k_j = 1, \forall j$) in the other sentence (Matusov et al., 2004; Taskar et al., 2005). Our matroid generalizations provide flexibility in modeling fertility, and also strategies for solving the word alignment problem efficiently and near-optimally. In particular, when f is monotone submodular, near-optimal solutions for Problem 1 can be efficiently guaranteed.

For example, in (Fisher et al., 1978), a simple greedy algorithm for monotone submodular function maximization with a matroid constraint is shown to have a constant approximation factor. Precisely, the greedy algorithm finds a solution A such that $f(A) \geq \frac{1}{m+1} f(A^*)$ where A^* is the optimal solution and m is number of matroid constraints. When there is only one matroid constraint, we get an approximation factor $\frac{1}{2}$. Constant factor approximation algorithms are particularly attractive since the quality of the solution does not depend on the size of the problem, so even very large size problems do well. It is also important to note that this is a worst case bound, and in most cases the quality of the solution obtained will be much better than this bound suggests.

Vondrák (2008) shows a continuous greedy algorithm followed by pipage rounding with approximation factor $1 - 1/e$ (≈ 0.63) for maximizing a monotone submodular function subject to a matroid constraint. Lee et al. (2009) improve the $\frac{1}{m+1}$-approximation result in (Fisher et al., 1978) by showing a local-search algorithm has approximation guarantee of $\frac{1}{m+\epsilon}$ for the problem of maximizing a monotone submodular function subject to m matroid constraints ($m \geq 2$ and $\epsilon > 0$). In this paper, however, we use the simple greedy algorithm for the sake of efficiency. We outline our greedy algorithm for Problem 1 in Algorithm 1, which is slightly different from the one in (Fisher et al., 1978) as in line 4 of Algorithm 1, we have an additional requirement on a such that the increment of adding a is *strictly* greater than zero. This additional requirement is to maintain a higher precision word alignment solution. The theoretical guarantee still holds as f is monotone — i.e., Algorithm 1 is a $\frac{1}{2}$-approximation algorithm for Problem 1 (only one matroid constraint) when f is monotone submodular.

Algorithm 1: A greedy algorithm for Problem 1.

```
   input : A = ∅, N = V.
1  begin
2      while N ≠ ∅ do
3          a ← argmax_{e∈N} f(A ∪ {e}) − f(A);
4          if A ∪ {a} ∈ I and f(A ∪ {a}) − f(A) > 0
           then
5              A → A ∪ {a}
6          N → N \ {a}.
7  end
```

Algorithm 1 requires $O(|V|^2)$ evaluations of f. In practice, the argmax in Algorithm 1 can be efficient implemented with priority queue when f is submodular (Minoux, 1978), which brings the complexity down to $O(|V| \log |V|)$ oracle function calls.

4 Submodular Fertility

We begin this section by demonstrating that submodularity arises naturally when modeling word fertility. To do so, we borrow an example of fertility from (Melamed, 2000). Suppose a trained model estimates $s(e_1, f_1) = .05$, $s(e_1, f_2) = .02$ and $s(e_2, f_2) = .01$, where $s(e_i, f_j)$ represents the score of aligning e_i and f_j. To find the correct alignment (e_1, f_1) and (e_2, f_2), the competitive linking algorithm in (Melamed, 2000) poses a one-to-one assumption to prevent choosing (e_1, f_2) over (e_2, f_2). The one-to-one assumption, however, limits the algorithm's capability of handling models with fertility larger than one. Alternatively, we argue that the reason of choosing (e_2, f_2) rather than (e_1, f_2) is that the benefit of aligning e_1 and f_2 *diminishes* after e_1 is already aligned with f_1 — this is exactly the property of diminishing returns, and therefore, it is natural to use submodular functions to model alignment scores.

To illustrate this further, we use another real example taken from the trial set of English-French Hansards data. The scores estimated from the data for aligning word pairs (*the, le*), (*the, de*) and (*of, de*) are $0.68, 0.60$ and 0.44 respectively. Given an English-French sentence pair: "*I have stressed the CDC as an example of creative, aggressive effective public ownership*" and "*je le ai cité comme exemple de propriété publique créatrice, dynamique et efficace*", an algorithm that allows word fertility larger than 1 might choose alignment (*the, de*) over (*of, de*) since $0.68 + 0.60 > 0.68 + 0.44$, regardless the fact that *the* is already aligned with *le*. Now if we use a submodular function to model the score of aligning an English word to a set of French words, we might obtain the correct alignments (*the, le*) and (*of, de*) by incorporating the diminishing returns property (i.e., the score gain of (*the, de*), which is 0.60 out of context, could diminish to something less than 0.44 when evaluated in the context of (*the, le*)).

Formally, for each i in E, we define a mapping

$\delta_i : 2^V \rightarrow 2^F$ with

$$\delta_i(A) = \{j \in F | (i,j) \in A\}, \quad (1)$$

i.e., $\delta_i(A)$ is the set of positions in F that are aligned with position i in alignment A.

We use function $f_i : 2^F \rightarrow \mathbb{R}_+$ to represent the benefit of aligning position $i \in E$ to a set of positions in F. Given score $s_{i,j}$ of aligning i and j, we could have, for $S \subseteq F$,

$$f_i(S) = \left(\sum_{j \in S} s_{i,j} \right)^{\alpha}, \quad (2)$$

where $0 < \alpha \leq 1$, i.e., we impose a concave function over a modular function, which produces a submodular function. The value of α determines the rate that the marginal benefit diminishes when aligning a word to more than one words in the other string.

Summing over alignment scores in all positions in E, we obtain the total score of an alignment A:

$$f(A) = \sum_{i \in E} f_i(\delta_i(A)), \quad (3)$$

which is again, monotone submodular. By diminishing the marginal benefits of aligning a word to more than one words in the other string, $f(A)$ encourages the common case of low fertility while allowing fertility larger than one. For instance in the aforementioned example, when $\alpha = \frac{1}{2}$, the score for aligning both *le* and *de* to *the* is $\sqrt{0.68 + 0.60} \approx 1.13$, while the score of aligning *the* to *le* and *of* to *de* is $\sqrt{0.68} + \sqrt{0.44} \approx 1.49$, leading to the correct alignment.

5 Experiments

We evaluated our approaches using the English-French Hansards data from the 2003 NAACL shared task (Mihalcea and Pedersen, 2003). This corpus consists of 1.1M automatically aligned sentences, and comes with a test set of 447 sentences, which have been hand-aligned and are marked with both "sure" and "possible" alignments (Och and Ney, 2003). Using these alignments, *alignment error rate* (AER) is calculated as:

$$AER(A, S, P) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \quad (4)$$

where S is the set of sure gold pairs, and P is the set of possible gold pairs. We followed the work in (Taskar et al., 2005) and split the original test set into 347 test examples, and 100 training examples for parameters tuning.

In general, the score of aligning i to j can be modeled as a function of arbitrary features. Although parameter learning in our framework would be another interesting topic to study, we focus herein on the inference problem. Therefore, only one feature (Eq. 5) was used in our experiments in order for no feature weight learning to be required. In particular, we estimated the score of aligning i to j as

$$s_{i,j} = \frac{p(f_j|e_i) \cdot p(i|j, I)}{\sum_{j' \in F} p(f_{j'}|e_i) \cdot p(i|j', I)}, \quad (5)$$

where the translation probability $p(f_j|e_i)$ and alignment probability $p(i|j, I)$ were obtained from IBM model 2 trained on the 1.1M sentences. The IBM 2 models gives an AER of 21.0% with French as the target, in line with the numbers reported in Och and Ney (2003) and Lacoste-Julien et al. (2006).

We tested two types of partition matroid constraints. The first is a global matroid constraint:

$$A \in \{A' \subseteq V : \forall j \in F, |A' \cap P_j^E| \leq b\}, \quad (6)$$

which restricts fertility of *all* words on F side to be at most b. This constraint is denoted as $\text{Fert}_F(A) \leq b$ in Table 1 for simplicity. The second type, denoted as $\text{Fert}_F(A) \leq k_j$, is word-dependent:

$$A \in \{A' \subseteq V : \forall j \in F, |A' \cap P_j^E| \leq k_j\}, \quad (7)$$

where the fertility of word on j is restricted to be at most k_j. Here $k_j = \max\{b : p_b(f) \leq \theta, b \in \{0, 1, \ldots, 5\}\}$, where θ is a threshold and $p_b(f)$ is the probability that French word f was aligned to at most b English words based on the IBM 2 alignment.

As mentioned in Section 3, matroid constraints generalize the matching constraint. In particular, when using two matroid constraints, $\text{Fert}_E(A) \leq 1$ and $\text{Fert}_F(A) \leq 1$, we have the matching constraint where fertility for both English and French words are restricted to be at most one. Our setup 1 (see Table 1) uses these two constraints along with a modular objective function, which is equivalent to the maximum weighted bipartite matching problem. Using

Table 1: AER results

ID	Objective function	Constraint	AER(%)
1	modular: $f(A) = \sum_{i \in E} \sum_{j \in \delta_i(A)} s_{i,j}$	$\text{Fert}_F(A) \leq 1, \text{Fert}_E(A) \leq 1$	21.0
2		$\text{Fert}_F(A) \leq 1$	23.1
3		$\text{Fert}_F(A) \leq k_j$	22.1
4	submodular: $f(A) = \sum_{i \in E} \left(\sum_{j \in \delta_i(A)} s_{i,j} \right)^{\alpha}$	$\text{Fert}_F(A) \leq 1$	19.8
5		$\text{Fert}_F(A) \leq k_j$	**18.6**
Generative model (IBM 2, E→F)			21.0
Maximum weighted bipartite matching			20.9
Matching with negative penalty on fertility (ILP)			19.3

greedy algorithm to solve this problem, we get AER 21.0% (setup 1 in Table 1) – no significant difference compared to the AER (20.9%) achieved by the exact solution (maximum weighted bipartite matching approach), illustrating that greedy solutions are near-optimal. Note that the bipartite matching approach does not improve performance over IBM 2 model, presumably because only one feature was used here.

When allowing fertility of English words to be more than one, we see a significant AER reduction using a submodular objective (setup 4 and 5) instead of a modular objective (setup 2 and 3), which verifies our claim that submodularity lends itself to modeling the marginal benefit of growing fertility. In setup 2 and 4, while allowing larger fertility for English words, we restrict the fertility of French words to be most one. To allow higher fertility for French words, one possible approach is to use constraint $\text{Fert}_F(A) \leq 2$, in which all French words are allowed to have fertility up to 2. This approach, however, results in a significant increase of false positive alignments since all French words tend to collect as many matches as permitted. This issue could be alleviated by introducing a symmetric version of the objective function in Eq. 3 such that marginal benefit of higher fertility of French words are also compressed. Alternatively, we use the second type of matroid constraint in which fertility upper bounds of French words are word-dependent instead of global. With $\theta = .8$, about 10 percent of the French words have k_j equal to 2 or greater. By using the word-dependent matroid constraint (setup 3 and 5), AERs are reduced compared to those using global matroid constraints. In particular, 18.6% AER is achieved by setup 5, which significantly outperforms the maximum weighted bipartite matching approach.

We also compare our method with model of Lacoste-Julien et al. (2006) which also allows fertility larger than one by penalizing different levels of fertility. We used $s_{i,j}$ as an edge feature and $p_b(f)$ as a node feature together with two additional features: a bias feature and the bucketed frequency of the word type. The same procedures for training and decoding as in (Lacoste-Julien et al., 2006) were performed where MOSEK was used as the ILP solver. As shown in Table 1, performance of setup 5 outperforms this model and moreover, our approach is at least 50 times faster: it took our approach only about half a second to align all the 347 test set sentence pairs whereas using the ILP-based approach took about 40 seconds.

6 Discussion

We have presented a novel framework where word alignment is framed as submodular maximization subject to matroid constraints. Our framework extends previous matching-based frameworks in two respects: submodular objective functions generalize modular (linear) objective functions, and matroid constraints generalize matching constraints. Moreover, such generalizations do not incur a prohibitive computational price since submodular maximization over matroids can be efficiently solved with performance guarantees. As it is possible to leverage richer forms of submodular functions that model higher order interactions, we believe that the full potential of our approach has yet to be explored. Our approach might lead to novel approaches for machine translation as well.

Acknowledgment

We thank Simon Lacoste-Julien for sharing his code and features from (Lacoste-Julien et al., 2006), and the anonymous reviewers for their comments. This work was supported by NSF award 0905341.

References

P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

J. Edmonds, 1970. *Combinatorial Structures and their Applications*, chapter Submodular functions, matroids and certain polyhedra, pages 69–87. Gordon and Breach.

ML Fisher, GL Nemhauser, and LA Wolsey. 1978. An analysis of approximations for maximizing submodular set functions—II. *Polyhedral combinatorics*, pages 73–87.

S. Jegelka and J. A. Bilmes. 2011. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, June.

D. Kempe, J. Kleinberg, and E. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th Conference on SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

V. Kolmogorov and R. Zabin. 2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

A. Krause and C. Guestrin. 2005. Near-optimal nonmyopic value of information in graphical models. In *Proc. of Uncertainty in AI*.

A. Krause, H.B. McMahan, C. Guestrin, and A. Gupta. 2008. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801.

S. Lacoste-Julien, B. Taskar, D. Klein, and M.I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 112–119. Association for Computational Linguistics.

J. Lee, M. Sviridenko, and J. Vondrák. 2009. Submodular maximization over multiple matroids via generalized exchange properties. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 244–257.

H. Lin and J. Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2010)*, Los Angeles, CA, June.

H. Lin and J. Bilmes. 2011. A class of submodular functions for document summarization. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, Portland, OR, June.

E. Matusov, R. Zens, and H. Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 219. Association for Computational Linguistics.

I.D. Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

R. Mihalcea and T. Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 1–10. Association for Computational Linguistics.

M. Minoux. 1978. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, pages 234–243.

Mukund Narasimhan and Jeff Bilmes. 2004. PAC-learning bounded tree-width graphical models. In *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI-2004)*. Morgan Kaufmann Publishers, July.

M. Narasimhan and J. Bilmes. 2005. A submodular-supermodular procedure with applications to discriminative structure learning. In *Proc. Conf. Uncertainty in Artifical Intelligence*, Edinburgh, Scotland, July. Morgan Kaufmann Publishers.

M. Narasimhan and J. Bilmes. 2007. Local search for balanced submodular clusterings. In *Twentieth International Joint Conference on Artificial Intelligence (IJCAI07)*, Hyderabad, India, January.

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

A. Schrijver. 2003. *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80. Association for Computational Linguistics.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

J. Vondrák. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 67–74. ACM.