Submodularity for Data Selection in Statistical Machine Translation

Katrin Kirchhoff
Department of Electrical Engineering
University of Washington
Seattle, WA, USA
kk2@u.washington.edu

Jeff Bilmes
Department of Electrical Engineering
University of Washington
Seattle, WA, USA
bilmes@u.washington.edu

Abstract

We introduce submodular optimization to the problem of training data subset selection for statistical machine translation (SMT). By explicitly formulating data selection as a submodular program, we obtain fast scalable selection algorithms with mathematical performance guarantees, resulting in a unified framework that clarifies existing approaches and also makes both new and many previous approaches easily accessible. We present a new class of submodular functions designed specifically for SMT and evaluate them on two different translation tasks. Our results show that our best submodular method significantly outperforms several baseline methods, including the widely-used cross-entropy based data selection method. In addition, our approach easily scales to large data sets and is applicable to other data selection problems in natural language processing.

1 Introduction

SMT has made significant progress over the last decade, not least due to the availability of increasingly larger data sets. Large-scale SMT systems are now routinely trained on millions of sentences of parallel data, and billions of words of monolingual data for language modeling. Large data sets are often beneficial, but they do create certain other problems. First, they place higher demands on computational resources (storage and compute). Hence, existing software infrastructure may need to be adapted and optimized to handle such large data sets. Second, experimental turn-around time is increased as well, making it more difficult to quickly train, fine-tune, and evaluate novel modeling approaches. Most importantly, however, SMT performance does not increase linearly with the training data size but levels off after a certain point. This is because the additional training data may be noisy, irrelevant to the task at hand, or inherently redundant. Thus, a linear increase in the amount of training data typically leads to a sublinear increase in performance, an effect known as diminishing returns. Several recent papers (Bloodgood and Callison-Burch, 2010; Turchi et al., 2012a; Turchi et al., 2012b) have amply demonstrated this effect.

A way to counteract this is to perform *data subset selection*, i.e., choose a subset of the available training data to optimize a particular quality criterion. One scheme is to select a subset that expresses as much of the information in the original data set as possible - i.e., the data set should be "summarized" by excluding redundant information. Another scheme, popular in the context of SMT, is to subselect the original training set to match the properties of a particular test set.

In this paper, we introduce submodularity for subselecting SMT training data, a methodology that follows both of the above schemes.[1] Submodular functions (Fujishige, 2005) are a class of discrete set functions having the property of diminishing returns. They occur naturally in a wide range of problems in a diverse set of fields including economics, game theory, operations research, circuit theory, and more recently machine learning. Submodular functions share certain properties with convexity (e.g., naturalness and mathematical tractability) although submodularity is still quite distinct from convexity.

We present a novel class of submodular functions particularly suited for SMT subselection and evaluate it against state-of-the-art baseline methods on two different translation tasks, showing that our method outperforms them significantly in most cases. While many approaches to SMT data selection have been developed previously (a detailed overview is provided in Section 3), many of them are heuristic and do not offer performance guarantees. Certain previous approaches, however, have

[1] As far as we know, submodularity has not before been explicitly utilized for SMT subset selection.

inadvertently made use of submodular methods. This, in addition to our own positive results, provides strong evidence that submodularity is a natural and practical framework for data subset selection in SMT and related fields.

An additional advantage of this framework is that many submodular programs (e.g., the greedy procedure reviewed in Section 2) are fast and scalable to large data sets. By contrast, trying to solve a submodular problem using, say, an integer-linear programming (ILP) procedure, would lead to impenetrable scalability problems.

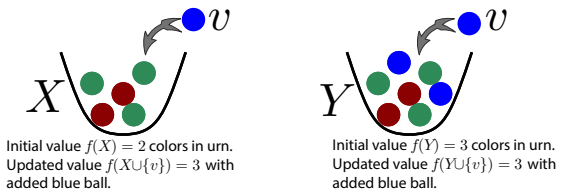

Figure 1: $f(Y)$ measures the number of distinct colors in the set of balls Y, and hence is submodular.

This paper makes several contributions: First, we present a brief overview of submodular functions (Section 2) and their potential application to natural language processing (NLP). Next we review previous approaches to MT data selection (Section 3) and analyze them with respect to their submodular properties. We find that some previous approaches are submodular in nature although this connection was not heretofore made explicit. Section 4 details our new approach. We discuss desirable properties of an SMT data selection objective and present a new class of submodular functions tailored towards this problem. Section 5 presents the data and systems used for the experiments, and results are reported in Section 6. Section 7 then concludes.

2 Submodular Functions/Optimization

Submodular functions (Edmonds, 1970; Fujishige, 2005), are widely used in mathematics, economics, circuit theory (Narayanan, 1997), and operations research. More recently, they have attracted much interest in machine learning (e.g., (Narasimhan and Bilmes, 2004; Kolmogorov and Zabih, 2004; Krause et al., 2008; Krause and Guestrin, 2011; Jegelka and Bilmes, 2011; Iyer and Bilmes, 2013)), where they have been applied to a variety of problems. In natural language and speech processing, they have been applied to document summarization (Lin and Bilmes, 2011; Lin and Bilmes, 2012) and speech data selection (Wei et al., 2013).

We are given a finite size-n set of objects V (i.e., $|V| = n$). A valuation function $f : 2^V \rightarrow \mathbb{R}_+$ is defined that returns a non-negative real value for any subset $X \subseteq V$. The function f is said to be *submodular* if it satisfies the property of diminishing returns: namely, for all $X \subseteq Y$ and $v \notin Y$, we must have:

$$f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y). \quad (1)$$

This means that the incremental value (or gain) of element v decreases when the context in which v is considered grows from X to $Y \supseteq X$. We define the "gain" as $f(v|X) \triangleq f(X \cup \{v\}) - f(X)$. Hence, f is submodular if $f(v|X) \geq f(v|Y)$. We note that a function $m : 2^V \rightarrow \mathbb{R}_+$ is said to be *modular* if it satisfies the above with equality, meaning $m(v|X) = m(v|Y)$ for all $X \subseteq Y \subseteq V \setminus \{v\}$. If m is modular and $m(\emptyset) = 0$, it can be written as $m(X) = \sum_{x \in X} m(x)$ and, moreover, is seen simply as a n-dimensional vector $m \in \mathbb{R}^V$.

As an example, suppose we have a set V of balls and $f(X)$ counts the number of colors present in any subset $X \subseteq V$. In Figure 1, $|X| = 5$ and $f(X) = 2$, $|Y| = 7$ and $f(Y) = 3$, and $X \subset Y$. Adding v (a blue ball) to X has a unity gain $f(v|X) = 1$ but since a blue ball exists in Y, we have $f(v|Y) = 0 < f(v|X) = 1$.

Submodularity is a natural model for data subset selection in SMT. In this case, each $v \in V$ is a distinct training data sentence and V corresponds to a training set. An important characteristic of any good model for this problem is that we wish to decrease the "value" of a sentence $v \in V$ based on how much that sentence has in common with those sentences, say X, that have already been chosen. The value $f(v|X)$ of a given sentence v in a context of previously chosen sentences $X \subseteq V$ further diminishes as the context grows $Y \supseteq X$. When, for example, a sentence's value is represented as the value of its set of features (e.g., n-grams), it is natural for those features' values to be discounted based on how much representation of those features already exists in a previously chosen subset. This corresponds to submodularity, which can easily be expressed mathematically by functions such as Eqn. (4) below.

Not only are submodular functions natural for SMT subset selection, they can also be optimized efficiently and scalably such that the result has mathematical performance guarantees. In the remainder of this paper we will assume that f is not only submodular, but also non-negative ($f(X) \geq 0$ for all X), and *monotone non-decreasing* ($f(X) \leq f(Y)$ for all $X \subseteq Y$). Such functions are trivial to uselessly maximize, since $f(V)$ is the largest possible valuation. Typically, however, we wish to have

Algorithm 1: The Greedy Algorithm

Input: Submodular function $f : 2^V \rightarrow \mathbb{R}_+$, cost vector m, budget b, finite set V.
Output: X_k where k is the number of iterations.
Set $X_0 \leftarrow \emptyset$; $i \leftarrow 0$;
while $m(X_i) < b$ **do**
    Choose v_i as follows:
    $v_i \in \left\{ \text{argmax}_{v \in V \setminus X_i} \frac{f(\{v\}|X_i)}{m(v)} \right\}$;
    $X_{i+1} \leftarrow X_i \cup \{v_i\}$; $i \leftarrow i+1$;

a valuable subset of bounded and small cost, where cost is measured based on a modular function $m(X)$. For example, the cost $m(v)$ of a sentence $v \in V$ might be its length, so $m(X) = \sum_{x \in X} m(x)$ is a sum of sentence lengths. This leads to the following optimization problem:

$$X^* \in \underset{X \subseteq V, m(X) \leq b}{\text{argmax}} \; f(X), \tag{2}$$

where b is a known budget. Solving this problem exactly is NP-complete (Feige, 1998), and expressing it as an ILP procedure renders it impractical for large data sizes. When f is submodular the cost is just size ($m(X) = |X|$), then the simple greedy algorithm (detailed below) will have a **worst-case guarantee** of $f(\tilde{X}^*) \geq (1 - 1/e) f(X_{\text{opt}}) \approx 0.63 f(X_{\text{opt}})$ where X_{opt} is the optimal and $\tilde{X}^*$ is the greedy solution (Nemhauser et al., 1978).

This constant factor guarantee has practical importance. First, a constant factor guarantee stays the same as n grows, so the relative worst-case quality of the solution is the same for small and for big problem instances. Second, the worst-case result is achieved only by very contrived and unrealistic function instances — the typical case is almost always much better. Third, the worst-case guarantee improves depending on the "curvature" $\kappa \in [0,1]$ of the submodular function (Conforti and Cornuejols, 1984). When the submodular function is not fully curved ($\kappa < 1$, something true of the functions used in this paper), the worst case guarantee is better, namely $\frac{1}{\kappa}(1 - e^{-\kappa})$ (e.g., a function f with $\kappa = 0.2$ has a worst-case guarantee of 0.91). Lastly, when the cost m is not just cardinality but an arbitrary non-negative modular function, a greedy algorithm has similar guarantees (Sviridenko, 2004), and a scalable variant has a worst-case guarantee of $1 - 1/\sqrt{e}$ (Lin and Bilmes, 2010).

The basic greedy algorithm has a very simple form. Starting with $X \leftarrow \emptyset$, we repeat the operation $X \leftarrow X \cup \text{argmax}_{v \in V \setminus X} \frac{f(v|X)}{m(v)}$ until the budget is exceeded ($m(X) > b$) and then backoff to the previous iteration (complete details are given in Algorithm 1). While the algorithm has complexity $O(n^2)$, there is an accelerated instance of this algorithm (Minoux, 1978; Leskovec et al., 2007) that has empirical computational complexity of $O(n \log n)$ where $n = |V|$. The greedy algorithm, therefore, scales practically to very large n. Recently, still much faster (Wei et al., 2014) and also parallel distributed (Mirzasoleiman et al., 2013) greedy procedures have been advanced offering still better scalability.

There are many submodular functions that are appropriate for subset selection (Lin and Bilmes, 2011; Lin and Bilmes, 2012). Some of them are graph-based, where we are given a non-negative weighted graph $G = (V, E, w)$ and $w : E \rightarrow \mathbb{R}_+$ is a set of edge weights (i.e., $w(x,y)$ is a non-negative similarity score between sentences x and y). A submodular function is obtained via a *graph cut* function $f(X) = \sum_{x \in X, y \in V \setminus X} w(x,y)$ or via a *monotone truncated graph cut* function $f(X) = \sum_{v \in V} \min(C_v(X), \alpha C_v(V))$ where $\alpha \in (0,1)$ is a scalar parameter and $C_v(X) = \sum_{x \in X} w(v,x)$ is a v-specific modular function. Alternatively, the class of *facility location functions* $f(X) = \sum_{v \in V} \max_{x \in X} w(x,v)$ have been widely and successfully used in the field of operations research, and are also applicable here.

In the worse case, the required graph construction has a worst-case complexity of $O(n^2)$. While sparse graphs can be used, this can be prohibitive when $n = |V|$ gets large. Another class of submodular functions that does not have this problem is based on a weighted *bipartite* graph $G = (V, U, E, w)$ where V are the left vertices, U are the right vertices, $E \subseteq V \times U$ is a set of edges, and $w : U \rightarrow \mathbb{R}_+$ is a set of non-negative weights on the vertices U. For $X \subseteq V$, the *bipartite neighborhood* function is defined as:

$$f(X) = w(\{u \in U : \exists x \in X \text{ with } (x,u) \in E\}) \tag{3}$$

This function is interesting for NLP applications since U can be seen as a set of "features" of the elements $v \in V$ (i.e., if V is a set of sentences, U can be the collective set of n-grams for multiple values of n, and $f(X)$ is the weight of the n-grams contained collectively in sentences X).[2] Given a set $X \subseteq V$,

[2] To be consistent with standard notation in previous literature, we overload the use of n in "n-grams" and the size of our set "$n = |V|$", even though the two ns have no relationship with each other.

we get value from the features of the elements $x \in X$, but we get credit for each feature only one time — once a given object $x \in X$ has a given feature $u \in U$, any additions to X by elements also having feature u offer no further credit via that feature.

Another interesting class of submodular functions, allowing additional credit from an element even when its features already exist in X, are what we call *feature-based* submodular functions. They involve sums of non-decreasing concave functions applied to modular functions (Stobbe and Krause, 2010) and take the following form:

$$f(X) = \sum_{u \in U} w_u \phi_u(m_u(X)) \quad (4)$$

where $w_u > 0$ is a *feature weight*, $m_u(X) = \sum_{x \in X} m_u(x)$ is a non-negative modular function specific to feature u, $m_u(x)$ is a *relevance score* (a non-negative scalar score indicating the relevance of feature u in object x), and ϕ_u is a u-specific non-negative non-decreasing *concave function*. The gain is $f(v|X) = \sum_{u \in U} w_u \Big(\phi(m_u(X \cup \{v\})) - \phi(m_u(X)) \Big)$, and thanks to ϕ_u's concavity, the term $\phi(m_u(X \cup \{v\})) - \phi(m_u(X))$ for each feature $u \in U$ is decaying as X grows. The rate of decay, and hence the degree of diminishing returns and ultimately the measure of redundancy of the information provided by the feature, is controlled by the concave function. The rate of decay is also related to the curvature κ of the submodular function (c.f. §2), with more aggressive decay having higher curvature (and a worse worst-case guarantee). The decay is a modeling choice that should be decided based on a given application.

Feature-based functions have the advantage that they do not require the construction of a pairwise graph; they have a cost of only $O(n|U|)$, which is **linear in the data size** and therefore scalable to large data set sizes.

We utilize this class for our subset selection experiments described in Section 4, where we use one global concave function $\phi_u = \phi$ for all $u \in U$. In this work we chose one particular set of features U. However, given the large body of research into NLP feature engineering (Jurafsky and Martin, 2009), this class is extensible beyond just this set, which makes it suitable for many other NLP applications.

Before describing our SMT-specific functions in detail, we review previous work on subset selection for SMT in the context of submodularity.

3 Previous Approaches

There have been many previous approaches to data subset selection in SMT. In this section, we show that some of them in fact correspond to submodular methods, thus introducing a connection between submodularity and the practical problem of SMT data selection. The fact that submodularity is implicitly and unintentionally used in previous work suggests that it is natural for this problem.

A currently widely-used data selection method in SMT (which we also use as a baseline in Section 6) uses the cross-entropy between two language models (Moore and Lewis, 2010), one trained on the test set of interest, and another trained on a large set of generic or out-of-domain training data. We call this the *cross-entropy method*. This method trains a test-set specific (or in-domain) language model, LM_{in}, and a generic (out-of- or mixed-domain) language model, LM_{out}. Each sentence $x \in V$ in the training data is given a probability score with both language models and then ranked in descending order based on the log ratio

$$m_{\text{ce}}(x) = \frac{1}{\ell(x)} \log[\Pr(x|\text{LM}_{\text{in}}) / \Pr(x|\text{LM}_{\text{out}})] \quad (5)$$

where $\ell(x)$ is the length of sentence x. Finally, the top N sentences are chosen. In (Axelrod et al., 2011) this method is extended to take both sides of the parallel corpus into account rather than just the source side. The cross-entropy approach values each sentence individually, without regard to any interaction with already selected sentences. This approach, therefore, is modular (a special case of submodular) and values a set X via $m(X) = \sum_{x \in X} m(x)$. Moreover, the thresholding method for choosing a subset corresponds exactly to the optimization problem in Eqn. (2) where $f \leftarrow m$ and the budget b is set to the sum of the top N sentence scores. Thanks to modularity, the problem is no longer NP-complete, and the threshold method solves Eqn. (2) exactly. On the other hand, a modular function does not have the diminishing returns property, and thus has no chance to represent interaction or redundancy between sentences. The chosen subset, therefore, might have an enormous overrepresentation of one aspect of the training data while having minimal or no representation of another aspect, a major vulnerability of this approach.

Other methods use information retrieval (Hildebrand et al., 2005; Lü et al., 2007) which can also be described as modular function optimization (e.g., take the top k scoring sentences). Duplicate

sentence removal is easily represented by a feature-based submodular function, Equation (4), where there is one sentence-specific feature per sentence and where $\phi_u(m_u(X)) = \min(|X \cap \{u\}|, 1)$ — once a sentence is chosen, its contribution is saturated so any duplicate sentence has a gain of zero. Also, the unseen n-gram function of (Eck et al., 2005; Bloodgood and Callison-Burch, 2010) corresponds to a *bipartite neighborhood* submodular function, with a weight function defined based on n-gram counts. Moreover their functions are optimized using the greedy algorithm; hence they in fact have a $1 - 1/e$ guarantee. Other methods have noted and dealt with the existence of redundancy in phrase-based systems (Ittycheriah and Roukos, 2007) by limiting the set of phrases — submodular optimization inherently removes redundancy. Also, (Callison-Burch et al., 2005; Lopez, 2007) involve modular functions but where selection is over subsets of phrases (rather than sentences as in our current work) and where multiple selections occur, each specific to an individual test set sentence rather than the entire test set.

In the feature-decay method, presented in (Biçici, 2011; Biçici and Yuret, 2011; Biçici, 2013), the value of a sentence is based on its decomposition into a set of feature values. As sentences are added to a set, the feature decay approach in general diminishes the value of each feature depending on how much of that feature has already been covered by those sentences previously chosen — the papers define a set of *feature decay functions* for this purpose.

Our analysis of (Biçici, 2011; Biçici and Yuret, 2011; Biçici, 2013), from the perspective of submodularity, has revealed an interesting connection. The feature decay functions used in these papers turn out to be derivatives of non-decreasing concave functions. For example, in one case $\phi'(a) = 1/(1+a)$ which is the derivative of the concave function $\phi(a) = \ln(1+a)$. We are given a constant initialization w_u for feature $u \in U$ — in the papers, they set either $w_u \leftarrow 1$, or $w_u \leftarrow \log(m(V)/m_u(V))$, or $w_u \leftarrow \log(m(V)/(1+m_u(V)))$, where $m(V) = \sum_u m_u(V)$, and where $m_u(X) = \sum_{x \in X} m_u(x)$ is the count of feature u within the set of sentences $X \subseteq V$. This yields the submodular feature function $f_u(X) = w_u \phi(m_u(X))$. The value of sentence v as measured by feature u in the context of X is the gain $f_u(v|X)$, which is a discrete derivative corresponding to $w_u/(1+m_u(X \cup \{v\}))$. An alternative decay function they define is given as $\phi'(a) = 1/(1+b^a)$ for a base b (they set $b \leftarrow 2$) which is the derivative of the following non-decreasing concave function:

$$\phi(a) = \left[1 - \frac{1}{\ln(b)} \ln\Big(1 + \exp\big(-a \ln(b)\big)\Big)\right] \quad (6)$$

We note that this function is saturating, meaning that it quickly reaches its asymptote at its maximum possible value. We can, once again, define a function specific for feature $u \in U$ as $f_u(X) = w_u \phi(m_u(X))$ with a gain $f_u(v|X)$ being a discrete derivative corresponding to $w_u/(1+b^{m_u(X \cup \{v\})})$.

The connection between this work and submodularity is not complete, however, without considering the method used for optimization. In fact, Algorithm 1 of (Biçici and Yuret, 2011) is precisely the accelerated greedy algorithm of (Minoux, 1978) applied to the submodular function corresponding to $f(X) = \sum_{u \in U} f_u(X)$, and Algorithm 1 of (Biçici, 2013) is the cost-normalized variant of this greedy algorithm corresponding to a knapsack constraint (Sviridenko, 2004). Thus, our analysis shows that these methods also have a $1 - 1/e$ performance guarantee and also the $O(n \log n)$ empirical complexity mentioned in Section 2. This is an important connection, as it furthers the evidence that submodularity is natural for the problem of SMT subset selection. This also increases the accessibility of this method since we may view it as a special case of Equation (4).

Another class of approaches focuses on active learning. In (Haffari et al., 2009) a large corpus of noisy parallel data is created automatically; a smaller set of samples is then selected from this set that receive human translations. A combination of several "informativeness" scores is computed on a sentence-level basis, and samples are selected via hierarchical adaptive sampling (Dasgupta and Hsu, 2008). In (Mandal et al., 2008) a measure of disagreement between different MT systems, as well as an entropy-based criterion are used to select additional data for annotation. In (Bloodgood and Callison-Burch, 2010) and (Ambati et al., 2010), active learning is combined with crowd-sourced annotations to produce large, human-translated data sets that are as informative as possible. In (Cao and Khudanpur, 2012), samples are selected for discriminative training of an MT system according to a greedy algorithm that tries to maximize overall quality. These methods address a different scenario (data selection for annotation or discriminative training) than the one considered here; however, we also note that the actual selection techniques employed in these papers do not appear to be submodular.

4 Novel Submodular Functions for SMT

In this section, we design a parameterized class of submodular functions useful for SMT training data subset selection. By staying within the realm of submodularity, we retain the advantages of the greedy algorithm, its theoretical performance assurances, and its scalability properties. At the same time this opens the door to a general framework for quickly exploring a much larger class of functions (with the same desirable properties) than before.

It is important to note that we are using submodularity as a "model" of the selection process, and the submodular objective acts as a surrogate for the actual SMT objective function. Thus, the mathematical guarantee we have is in terms of the surrogate objective rather than the true SMT objective. Evaluating one point of the actual SMT objective would require the complete training and testing of an SMT system, so even an algorithm as efficient as Algorithm 1 would be infeasible, even on small data. It is therefore important to design a natural and scalable surrogate objective.

We do not consider the graph-based functions discussed in Section 2 here since they require a pairwise similarity matrix over all training sentences and thus have $O(n^2)$ worst-case complexity. For large tasks with millions or even billions of sentences, this eventually becomes impractical. Instead we focus on feature-based functions of the type presented in Eqn. (4), where each sentence is represented as a set of features rather than as a vertex in a graph. In this function there are four components to specify: 1) U, the *linguistic feature set*; 2) $m_u(x)$, the *relevance scores* for each feature u and sentence x; 3) w_u, the *feature weights*; and 4) ϕ, the *concave function* (we use one concave function, so $\phi_u = \phi$ for all $u \in U$).

Feature set: U is the set of n-grams from either the source language U^{src}, or from both the source and target language $U^{\text{src}} \cup U^{\text{tgt}}$ (see Section 6); since we are interested in selecting a training set that matches a given test set, we use the set of n-grams that occur both in the training set and in the development/test data (for target features, only development set features are used). I.e., $U^{\text{src}} = (U^{\text{src}}_{\text{dev}} \cup U^{\text{src}}_{\text{test}}) \cap U^{\text{src}}_{\text{train}}$ and $U^{\text{tgt}} = U^{\text{tgt}}_{\text{dev}} \cap U^{\text{tgt}}_{\text{train}}$.

Relevance scores: A feature u within a sentence x should be valued based on how salient that feature is within the "document" in which it occurs; here, the "document" is the set of training sentences. This is a task well suited to TFIDF. As an alternative to raw feature counts we thus also consider scores of the form $m_u(x) \leftarrow \text{tfidf}(u,x) = \text{tf}(u,x) \times \text{idf}^{\text{trn}}(u)$, where $\text{tf}(u,x)$ and $\text{idf}^{\text{trn}}(u)$ are defined as usual.

Feature weights: We wish to select those training samples that contain features occurring frequently in the test data while avoiding the over-selection of features that are very frequent in the training data because those are likely to be translated correctly anyway. This is similar to the approach in (Moore and Lewis, 2010) (see Equation (5)), where a log-probability ratio of in-domain to out-of-domain language model is utilized. In the present case, we need a value that is specific to feature $u \in U$; a natural approach is to use the ratio of counts $c^{\text{tst}}(u)/c^{\text{trn}}(u)$ where $c^{\text{tst}}(u)$ is the raw count of u in the development/test data, and $c^{\text{trn}}(u)$ is its raw count in the training data (note that $c^{\text{trn}}(u)$ is never zero due to the way U is defined). As an additional factor we allow feature length to have an influence. In general, longer n-grams might be considered more valuable since they typically lead to better translations and are more relevant for BLEU. Thus, we include a reward term for longer n-grams in the form of $\beta^{|u|}$ where $\beta \geq 1$ and $|u|$ is the length of feature u. This gives greater weight to longer n-grams when $\beta > 1$.

Concave function: It is imperative to find the right form of concave function since, as described in Section 2, the concave shape determines the degree to which redundancy and diminishing returns are represented. Intuitively, when the shape of the concave function for a feature becomes "flat" rapidly, that feature quickly looses its ability to provide additional value to a candidate subset. Many different concave functions were tested for ϕ, including one of the two functions implicit in (Biçici and Yuret, 2011) and derived in Section 3, and a variety of roots of the form $\phi(a) = a^{\alpha}$ for $0 < \alpha < 1$. In Table 2, for example, we find evidence that the simple square root $\phi(a) = \sqrt{a}$ performs slightly better than the log function. The square root is much less curved and decays much more gradually than either of the two functions implicit in (Biçici and Yuret, 2011), of which one is a log form and the other is even more curved and quickly saturates (see §3). The square root function yields a less curved submodular function, in the sense of (Conforti and Cornuejols, 1984), resulting in better worst-case guarantees. Indeed, Table 1 in (Biçici and Yuret, 2011) corroborates by showing that the more curved saturating function does worse than the less curved log function.

Four Components Together: Different instantia-

tions of the four components discussed above will result in different submodular functions of the general class defined in Eqn. (4). Particular settings of these general parameters produce the methods considered in (Biçici and Yuret, 2011), thus making that approach easily accessible once the general submodular framework is set up. As a very special case, this is also true of the cross-entropy method (Moore and Lewis, 2010), where $|U| = 1$, $m_u(x) \leftarrow \exp(m_{\text{ce}}(x))$ of Equation (5) [3], $w_u \leftarrow 1$, and $\phi(a) = a$ is the identity function. In Section 6, we specify the parameter settings used in our experiments.

Task	Train	Dev	Test	LM
NIST	189M	48k	49k	2.5B
Europarl	52.8M	57.7k	58.1k	53M

Table 1: Data set sizes (number of source-side words) for MT tasks. LM = language model data.

5 Data and Systems

We evaluate our approach on the NIST Arabic-English translation task, using the NIST 2006 set for development and the NIST 2009 set for evaluation. The training data consists of all Modern Standard Arabic-English parallel LDC corpora permitted in the NIST evaluations (minus the restricted time periods). Together these sets form a mixed-domain training set containing relevant in-domain data similar to the NIST data sets but also less relevant data (e.g., the UN parallel corpora); we thus expect data selection to work well on this task. Additional English language modeling data was drawn from several other LDC corpora (English Gigaword, AQUAINT, HARD, ANC/DCI and the American National Corpus). Preprocessing included conversion of the Arabic data to Buckwalter format, tokenization, spelling normalization, and morphological segmentation using MADA (Habash et al., 2009). Numbers and URLs were replaced with variables. The English data was tokenized and lowercased. Postprocessing involved recasing the translation output, replacing variable names with their original corresponding tokens, and normalizing spelling and stray punctuation marks. The recasing model is an SMT system without reordering, trained on parallel cased and lowercased versions of the training data. The recasing model remains fixed for all experiments and is not retrained for different sizes of the training data. Evaluation follows the NIST guidelines and was done by computing BLEU scores using the official NIST evaluation tool mteval-v13a.pl with the $-c$ flag for case-sensitive scoring. In addition to the NIST task we also applied our method to the Europarl German-English translation task. The training data comes from the Europarl-v7 collection[4]; the development set is the 2006 dev set, and the test set is the 2007 test set. The number of reference translations is 1. The German data was preprocessed by tokenization, lower-casing, splitting noun compounds and lemmatization to address morphological variation in German. The English side was tokenized and lowercased. Evaluation was done by computing BLEU on the lowercased versions of the data. Since test and training data for this task come from largely the same domain we expect the training data to be less redundant or irrelevant; nevertheless it will be interesting to see how much different data selection methods can contribute even to in-domain translation tasks. The sizes of the various data sets are shown in Table 1.

All translation systems were trained using the GIZA++/Moses infrastructure (Koehn et al., 2007). The translation model is a standard phrase-based model with a maximum phrase length of 7. Since a large number of experiments had to be run for this study, more complex hierarchical or syntax-based translation models were deliberately excluded in order to limit the experimental turn-around time needed for each experiment. The reordering model is a hierarchical model according to (Galley and Manning, 2008). The feature weights in the log-linear function were optimized on the development set BLEU score using minimum error-rate training. The language models for the NIST task (5-grams) were trained on three different data sources (Gigaword, GALE data, and all remaining corpora), which were then interpolated into a single model. The interpolation weights were optimized separately for the two different genres present in the NIST task (newswire and web text). All models used Witten-Bell discounting and interpolation of higher-order and lower-order models. Language models remain fixed for all experiments, i.e., the language model training data is not subselected since we were interested in the effect of data subset selection on the translation model only. The language model for the Europarl system was a 5-gram trained on Europarl data only.

[3] Due to modularity, any monotone increasing transformation from $m_{\text{ce}}(x)$ to $m_u(x)$ that ensures $m_u(x) \geq 0$ is equivalent.

[4] http://http://www.statmt.org/europarl/

Method	Data Subset Sizes			
	10%	20%	30%	40%
Rand	0.3991 (± 0.004)	0.4142 (± 0.003)	0.4205 (± 0.002)	0.4220 (± 0.002)
Xent	0.4235 (± 0.004)	0.4292 (± 0.002)	0.4290 (± 0.003)	0.4292 (± 0.001)
SM-1	**0.4309** (± 0.000)	**0.4367** (± 0.001)	0.4330 (± 0.004)	**0.4351** (± 0.002)
SM-2	**0.4330*** (± 0.001)	**0.4395*** (± 0.003)	0.4333 (± 0.001)	**0.4366*** (± 0.003)
SM-3	**0.4313*** (± 0.002)	**0.4338** (± 0.002)	**0.4361*** (± 0.002)	**0.4351** (± 0.003)
SM-4	0.4276 (± 0.003)	0.4303 (± 0.002)	0.4324 (± 0.002)	0.4329 (± 0.000)
SM-5	0.4285 (± 0.004)	**0.4356** (± 0.002)	0.4333 (± 0.003)	**0.4324** (± 0.002)
SM-6	**0.4302*** (± 0.004)	**0.4334** (± 0.003)	**0.4371*** (± 0.002)	**0.4349** (± 0.003)
SM-7	**0.4295** (± 0.002)	**0.4374** (± 0.002)	**0.4344** (± 0.001)	0.4314 (± 0.0004)
SM-8	**0.4304*** (± 0.002)	**0.4323** (± 0.000)	**0.4358** (± 0.003)	0.4337 (± 0.001)
100%	0.4257			

Table 2: BLEU scores (standard deviations) on the NIST 2009 (Ara-En) test set for random (Rand), cross-entropy (Xent), and submodular (SM) data selection methods defined in Table 4. 100% = system using all of the training data. Boldface numbers indicate a statistically significant improvement ($p \leq 0.05$) over the median Xent system. Starred scores are also significantly better than SM-5.

Method	Data Subset Sizes			
	10%	20%	30%	40%
Rand	0.2590 (± 0.003)	0.2652 (± 0.001)	0.2677 (± 0.002)	0.2697 (± 0.001)
Xent	0.2639 (± 0.002)	0.2687 (± 0.002)	0.2704 (± 0.001)	0.2723 (± 0.001)
SM-5	0.2653 (± 0.001)	**0.2727** (± 0.000)	0.2697 (± 0.002)	0.2720 (± 0.002)
SM-6	**0.2697**$*$ (± 0.001)	**0.2700** (± 0.002)	**0.2740**$*$ (± 0.002)	0.2723 (± 0.000)
100%	0.2651			

Table 3: BLEU scores (standard deviation) on the Europarl translation task for random (Rand), cross-entropy (Xent), and submodular (SM) data selection methods. 100% = system using all of the training data. Boldface numbers indicate a statistically significant improvement ($p \leq 0.05$) over the median Xent system. Starred scores are significantly better than SM-5.

6 Experiments

	Function parameters			
	$w(u)$	$\phi(a)$	$m_u(x)$	U
SM-1	$\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}\beta^{\lvert u\rvert}$	$\sqrt{a}$	tfidf(u,x)	U^{src}
SM-2	$\sqrt{\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}\beta^{\lvert u\rvert}}$	$\sqrt{a}$	tfidf(u,x)	$U^{\text{src}} \cup U^{\text{tgt}}$
SM-3	$\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}\beta^{\lvert u\rvert}$	$\sqrt{a}$	c(u,x)	U^{src}
SM-4	$c^{\text{tst}}(u)$	$\sqrt{a}$	tfidf(u,x)	U^{src}
SM-5	1	$ln(1+a)$	c(u,x)	U^{src}
SM-6	$\sqrt{\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}}$	$\sqrt{a}$	tfidf(u,x)	U^{src}
SM-7	$\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}$	$\sqrt{a}$	tfidf(u,x)	$U^{\text{src}} \cup U^{\text{tgt}}$
SM-8	$\frac{c^{\text{tst}}(u)}{c^{\text{trn}}(u)}$	$ln(1+a)$	tfidf(u,x)	$U^{\text{src}} \cup U^{\text{tgt}}$

Table 4: Different instantiations of the general submodular function in Eq. 4 ($\beta = 1.5$ in all cases).

We first trained a baseline system on 100% of the training data. Different data selection methods were then used to select subsets of 10%, 20%, 30% and 40% of the data. While not reported in the tables, above 40%, the performance slowly drops to the 100% performance.

The first baseline selection method utilizes random data selection, for which 3 different data sets of the specified size were drawn randomly from the training data. Individual systems were trained on all random subsets of the same size, and their scores were averaged. The second baseline is the cross-entropy method by (Moore and Lewis, 2010). In-domain language models were trained on the combined development and test data, and out-of-domain models were trained on an equivalent amount of data drawn randomly from the training set. Sentences were ranked by the function in Eq. 5, and the top k percent were chosen. The order of the n-gram models was optimized on the development set and was found to be 3. Larger model orders resulted in worse performance, possibly due to the

limited size of the data used for their training. Since this method also involves random data selection, we report the average BLEU score over 5 different trials. For the submodular selection method, Table 4 shows the different values that were tested for the four components listed in Section 4. The combination was optimized on the development set. The selection algorithm (Alg. 1) runs within a few minutes on our complete training set of 189M words.

Results on the NIST 2009 test set are shown in Table 2. The scores for the submodular systems are averages over 3 different runs of MERT tuning. Random data subset selection (Row 1) falls short of the baseline system using 100% of the training data. The cross-entropy method (Row 2) surpasses the performance of the baseline system at about 20% of the data, demonstrating that data subset selection is a suitable technique for such mixed-domain translation tasks. The following rows show results for the various submodular functions shown in Table 4. Out of these, SM-5 corresponds to the best approach in (Biçici and Yuret, 2011). SM-6 is our own best-performing function, beating the cross-entropy method by a statistically significant margin ($p \leq 0.05$) under all conditions.[5] SM-6 is also significantly better than SM-5 in two cases. Finally, it surpasses the performance of the all-data system at only 10% of the training data; possibly, even smaller training data sets could be used but this option was not investigated. While the bilingual submodular functions SM-2 and SM-7) yield an improvement of up to 0.015 BLEU points on the dev set (not shown in the table), they do not consistently outperform the monolingual functions on the test set. Since test set target features cannot be used in our scenario, bilingual features are only helpful to the extent that the development set closely matches the test set. However, target features should be quite helpful when selecting data from an out-of-domain set to match an in-domain training set (as in e.g. (Axelrod et al., 2011)). We found no gain from the length reward $\beta^{|u|}$.

The Europarl results (Table 3) show a similar pattern. Although the differences in BLEU scores are smaller overall (as expected on an in-domain translation task), data subset selection improves over the all-data baseline system in this case as well. The cross-entropy method again outperforms random data selection. On this task we only tested our submodular function that worked best on the NIST task; again we find that it outperforms the cross-entropy method. In two conditions (10% and 30%) these differences are statistically significant. 10% of the training data suffices to outperform the all-data system, and up to a full BLEU point can be gained on this task using 20-30% of the data and a submodular data selection method.

[5] Statistical significance was measured using the paired bootstrap resampling test of (Koehn, 2004), applied to the systems with the median BLEU scores.

7 Conclusions

We have introduced submodularity to SMT data subset selection, generalizing previous approaches to this problem. Our method has theoretical performance guarantees, comes with scalable algorithms, and significantly improves over current, widely-used data selection methods on two different translation tasks. There are many possible extensions to this work. One strategy would be to extend the feature set U with features representing different types of linguistic information - e.g., when using a syntax-based system it might be advantageous to select training data that covers the set of syntactic structures seen in the test data. Secondly, the selected data was test data specific. In some contexts, it is not possible to train test data specific systems dynamically; in that case, different submodular functions could be designed to select a representative "summary" of the training data. Finally, the use of submodular functions for subset selection is applicable to other data sets that can be represented as features or as a pairwise similarity graph. Submodularity thus can be applied to a wide range of problems in NLP beyond machine translation.

Acknowledgments

This material is based on research sponsored by Intelligence Advanced Research Projects Activity (IARPA) under agreement number FA8650-12-2-7263, and is also supported by the National Science Foundation under Grant No. (IIS-1162606), and by a Google, a Microsoft, and an Intel research award. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Intelligence Advanced Research Projects Activity (IARPA) or the U.S. Government.

References

[Ambati et al.2010] V. Ambati, S. Vogel, and J. Carbonell. 2010. Active learning and crowd-sourcing for machine translation. In *Proceedings of LREC*, pages 2169–2174, Valletta, Malta.

[Axelrod et al.2011] A. Axelrod, X. He, and J. Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP*, pages 355–362, Edinburgh, Scotland.

[Biçici and Yuret2011] E. Biçici and D. Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 272–283.

[Biçici2013] E. Biçici. 2013. Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 78–84.

[Biçici2011] E. Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, KOÇ University.

[Bloodgood and Callison-Burch2010] M. Bloodgood and C. Callison-Burch. 2010. Bucking the trend: large-scale cost-focused active learning for statistical machine translation. In *Proceedings of ACL*, pages 854–864.

[Callison-Burch et al.2005] Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 255–262. Association for Computational Linguistics.

[Cao and Khudanpur2012] Y. Cao and S. Khudanpur. 2012. Sample selection for large-scale MT discriminative training. In *Proceedings of AMTA*.

[Conforti and Cornuejols1984] M. Conforti and G. Cornuejols. 1984. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274.

[Dasgupta and Hsu2008] S. Dasgupta and D. Hsu. 2008. Hierarchical sampling for active learning. In *Proceedings of ICML*.

[Eck et al.2005] M. Eck, S. Vogel, and A. Waibel. 2005. Low cost portability for statistical machine translation based on n-gram frequency and tf-idf. In *Proceedings of the 10th Machine Translation Summit X*, pages 227–234.

[Edmonds1970] J. Edmonds, 1970. *Combinatorial Structures and their Applications*, chapter Submodular functions, matroids and certain polyhedra, pages 69–87. Gordon and Breach.

[Feige1998] U. Feige. 1998. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652.

[Fujishige2005] S. Fujishige. 2005. *Submodular functions and optimization. Annals of Discrete Mathematics*, volume 58. Elsevier Science.

[Galley and Manning2008] M. Galley and C. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP*, pages 847–855.

[Habash et al.2009] N. Habash, O. Rambow, and R. Roth. 2009. A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the MEDAR conference*, pages 102–109.

[Haffari et al.2009] G. Haffari, M. Roy, and A. Sarkar. 2009. Active learning for statistical machine translation. In *Proceedings of HLT*, pages 415–423.

[Hildebrand et al.2005] A. Hildebrand, M. Eck, S. Vogel, and A. Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT*, pages 133–142.

[Ittycheriah and Roukos2007] A. Ittycheriah and S. Roukos. 2007. Direct translation model 2. In *Proceedings of HLT/NAACL*, page 5764.

[Iyer and Bilmes2013] R. Iyer and J. Bilmes. 2013. Submodular optimization with submodular cover and submodular knapsack constraints. In *Neural Information Processing Society (NIPS)*, Lake Tahoe, CA, December.

[Jegelka and Bilmes2011] Stefanie Jegelka and Jeff A. Bilmes. 2011. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, June.

[Jurafsky and Martin2009] D. Jurafsky and J. H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, 2nd edition.

[Koehn et al.2007] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.

[Koehn2004] P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

[Kolmogorov and Zabih2004] V. Kolmogorov and R. Zabih. 2004. What energy functions can be minimized via graph cuts? *IEEE TPAMI*, 26(2):147–159.

[Krause and Guestrin2011] A. Krause and C. Guestrin. 2011. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4).

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

[Leskovec et al.2007] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429.

[Lin and Bilmes2010] H. Lin and J. Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of NAACL-HLT*, pages 2761–2801.

[Lin and Bilmes2011] H. Lin and J. Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of ACL*, pages 510–520.

[Lin and Bilmes2012] H. Lin and J. Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *Uncertainty in Artifical Intelligence (UAI)*, Catalina Island, USA, July. AUAI.

[Lopez2007] A. Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *EMNLP-CoNLL*, pages 976–985.

[Lü et al.2007] Y. Lü, J. Huang, and Q. Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of EMNLP*, pages 343–350.

[Mandal et al.2008] A. Mandal, D. Vergyri, W. Wang, J. Zheng, A. Stolcke, D. Hakkani-Tür G. Tür, and N.F. Ayan. 2008. Efficient data selection for machine translation. In *Proceedings of the Spoken Language Technology Workshop*, pages 261–264.

[Minoux1978] M. Minoux. 1978. Accelerated greedy algorithms for maximizing submodular functions. In *Lecture Notes in Control and Information Sciences*, volume 7, pages 234–243.

[Mirzasoleiman et al.2013] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. 2013. Distributed submodular maximization: Identifying representative elements in massive data. In *Neural Information Processing Systems (NIPS)*.

[Moore and Lewis2010] R. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the Association for Computational Linguistics*, pages 220–224.

[Narasimhan and Bilmes2004] M. Narasimhan and J. 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.

[Narayanan1997] H. Narayanan. 1997. Submodular functions and electrical networks. *Elsevier*.

[Nemhauser et al.1978] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions i. *Mathematical Programming*, 14:265–294.

[Stobbe and Krause2010] P. Stobbe and A. Krause. 2010. Efficient minimization of decomposable submodular functions. In *NIPS*.

[Sviridenko2004] M. Sviridenko. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43.

[Turchi et al.2012a] M. Turchi, T. de Bie, C. Goutte, and N. Cristianini. 2012a. Learning to translate: A statistical and computational analysis. *Advances in Artificial Intellligence*, 2012:484580:15 pages.

[Turchi et al.2012b] M. Turchi, C. Goutte, and N. Cristianini. 2012b. Learning machine translation from in-domain and out-of-domain data. In *Proceedings of EAMT*, Trento, Italy.

[Wei et al.2013] K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes. 2013. Using document summarization techniques for speech data subset selection. In *Proceedings of NAACL*, pages 721–726, Atlanta, Georgia, June.

[Wei et al.2014] K. Wei, R. Iyer, and Jeff Bilmes. 2014. Fast multi-stage submodular maximization. In *Proceedings of ICML*, Beijing, China.