Optimal Data Set Selection: An Application to Grapheme-to-Phoneme Conversion

Young-Bum Kim and Benjamin Snyder
University of Wisconsin-Madison
{ybkim,bsnyder}@cs.wisc.edu

Abstract

In this paper we introduce the task of unlabeled, optimal, data set selection. Given a large pool of unlabeled examples, our goal is to select a small subset to label, which will yield a high performance supervised model over the entire data set. Our first proposed method, based on the rank-revealing QR matrix factorization, selects a subset of words which span the entire word-space effectively. For our second method, we develop the concept of *feature coverage* which we optimize with a greedy algorithm. We apply these methods to the task of grapheme-to-phoneme prediction. Experiments over a data-set of 8 languages show that in all scenarios, our selection methods are effective at yielding a small, but optimal set of labelled examples. When fed into a state-of-the-art supervised model for grapheme-to-phoneme prediction, our methods yield average error reductions of 20% over randomly selected examples.

1 Introduction

Over the last 15 years, supervised statistical learning has become the dominant paradigm for building natural language technologies. While the accuracy of supervised models can be high, expertly annotated data sets exist for a small fraction of possible tasks, genres, and languages. The would-be tool builder is thus often faced with the prospect of annotating data, using crowd-sourcing or domain experts. With limited time and budget, the amount of data to be annotated might be small, especially in the prototyping stage, when the exact specification of the prediction task may still be in flux, and rapid prototypes are desired.

In this paper, we propose the problem of unsupervised, optimal data set selection. Formally, given a large set $\mathcal{X}$ of n unlabeled examples, we must select a subset $\mathcal{S} \subset \mathcal{X}$ of size $k \ll n$ to label. Our goal is to select such a subset which, when labeled, will yield a high performance supervised model over the entire data set $\mathcal{X}$. This task can be thought of as a zero-stage version of active learning: we must choose a single batch of examples to label, without the benefit of any prior labelled data points. This problem definition avoids the practical complexity of the active learning set-up (many iterations of learning and labeling), and ensures that the labeled examples are not tied to one particular model class or task, a well-known danger of active learning (Settles, 2010). Alternatively, our methods may be used to create the initial seed set for the active learner.

Our initial testbed for optimal data set selection is the task of grapheme-to-phoneme conversion. In this task, we are given an out-of-vocabulary word, with the goal of predicting a sequence of phonemes corresponding to its pronunciation. As training data, we are given a pronunciation dictionary listing words alongside corresponding sequences of phones, representing canonical pronunciations of those words. Such dictionaries are used as the final bridge between written and spoken language for technologies that span this divide, such as speech recognition, text-to-speech generation, and speech-to-speech language translation. These dictionaries are necessary: the pronunciation of words

Proceedings of NAACL-HLT 2013, pages 1196–1205,
Atlanta, Georgia, 9–14 June 2013.

continues to evolve after their written form has been fixed, leading to a large number of rules and irregularities. While large pronunciation dictionaries of over 100,000 words exist for several major languages, these resources are entirely lacking for the majority of the world's languages. Our goal is to automatically select a small but optimal subset of words to be annotated with pronunciation data.

The main intuition behind our approach is that the subset of selected data points should efficiently cover the range of phenomena most commonly observed across the pool of unlabeled examples. We consider two methods. The first comes from a line of research initiated by the numerical linear algebra community (Golub, 1965) and taken up by computer science theoreticians (Boutsidis et al., 2009), with the name COLUMN SUBSET SELECTION PROBLEM (CSSP). Given a matrix A, the goal of CSSP is to select a subset of k columns whose span most closely captures the range of the full matrix. In particular, the matrix $\tilde{A}$ formed by orthogonally projecting A onto the k-dimensional space spanned by the selected columns should be a good approximation to A. By defining A^T to be our data matrix, whose rows correspond to words and whose columns correspond to features (character 4-grams), we can apply the CSSP randomized algorithm of (Boutsidis et al., 2009) on A to obtain a subset of k words which best span the entire space of words.

Our second approach is based on a notion of *feature coverage*. We assume that the benefit of seeing a feature f in a selected word bears some positive relationship to the frequency of f in the unlabeled pool. However, we further assume that the lion's share of benefit accrues the first few times that we label a word with feature f, with the marginal utility quickly tapering off as more such examples have been labeled. We formalize this notion and provide an exact greedy algorithm for selecting the k data points with maximal feature coverage.

To assess the benefit of these methods, we apply them to a suite of 8 languages with pronunciation dictionaries. We consider ranges from 500 to 2000 selected words and train a start-of-the-art grapheme-to-phoneme prediction model (Bisani and Ney, 2008). Our experiments show that both methods produce significant improvements in prediction quality over randomly selected words, with our feature coverage method consistently outperforming the randomized CSSP algorithm. Over the 8 languages, our method produces average reductions in error of 20%.

2 Background

Grapheme-to-phoneme Prediction The task of grapheme-to-phoneme conversion has been considered in a variety of frameworks, including neural networks (Sejnowski and Rosenberg, 1987), rule-based FSA's (Kaplan and Kay, 1994), and pronunciation by analogy (Marchand and Damper, 2000). Our goal here is not to compare these methods, so we focus on the probabilistic joint-sequence model of Bisani and Ney (2008). This model defines a joint distribution over a grapheme sequence $\boldsymbol{g} \in G^*$ and a phoneme sequence $\boldsymbol{\phi} \in \Phi^*$, by way of an unobserved *co-segmentation* sequence $\boldsymbol{q}$. Each co-segmentation unit q_i is called a *graphone* and consists of an aligned pair of zero or one graphemes and zero or one phonemes: $q_i \in G \cup \{\epsilon\} \times \Phi \cup \{\epsilon\}$.[1] The probability of a joint grapheme-phoneme sequence is then obtained by summing over all possible co-segmentations:

$$P(\boldsymbol{g}, \boldsymbol{\phi}) = \sum_{\boldsymbol{q} \in S(\boldsymbol{g}, \boldsymbol{\phi})} P(\boldsymbol{q})$$

where $S(\boldsymbol{g}, \boldsymbol{\phi})$ denotes the set of all graphone sequences which yield $\boldsymbol{g}$ and $\boldsymbol{\phi}$. The probability of a graphone sequence of length K is defined using an h-order Markov model with multinomial transitions:

$$P(\boldsymbol{q}) = \prod_{i=1}^{k+1} P(q_i | q_{i-h}, \ldots, q_{i-1})$$

where special start and end symbols are assumed for $q_{j<1}$ and q_{k+1}, respectively.

To deal with the unobserved co-segmentation sequences, the authors develop an EM training regime that avoids overfitting using a variety of smoothing and initialization techniques. Their model produces state-of-the-art or comparable accuracies across a

[1] The model generalizes easily to graphones consisting of more than one grapheme or phoneme, but in both (Bisani and Ney, 2008) and our initial experiments we found that the 01-to-01 model always performed best.

wide range of languages and data sets.[2] We use the publicly available code provided by the authors.[3] In all our experiments we set $h = 4$ (i.e. a 5-gram model), as we found that accuracy tended to be flat for $h > 4$.

Active Learning for G2P Perhaps most closely related to our work are the papers of Kominek and Black (2006) and Dwyer and Kondrak (2009), both of which use active learning to efficiently bootstrap pronunciation dictionaries. In the former, the authors develop an active learning word selection strategy for inducing pronunciation rules. In fact, their greedy n-gram selection strategy shares some of the some intuition as our second data set selection method, but they were unable to achieve any accuracy gains over randomly selected words without active learning.

Dwyer and Kondrak use a Query-by-Bagging active learning strategy over decision tree learners. They find that their active learning strategy produces higher accuracy across 5 of the 6 languages that they explored (English being the exception). They extract further performance gains through various refinements to their model. Even so, we found that the Bisani and Ney grapheme-to-phoneme (G2P) model (Bisani and Ney, 2008) always achieved higher accuracy, even when trained on random words. Furthermore, the relative gains that we observe using our optimal data set selection strategies (without any active learning) are much larger than the relative gains of active learning found in their study.

Data Set Selection and Active Learning Eck et al (2005) developed a method for training compact Machine Translation systems by selecting a subset of sentences with high n-gram coverage. Their selection criterion essentially corresponds to our feature coverage selection method using coverage function cov_2 (see Section 3.2). As our results will show, the use of a geometric feature discount (cov_3) provided better results in our task.

Otherwise, we are not aware of previous work proposing optimal data set selection as a general research problem. Of course, active learning strategies can be employed for this task by starting with a small random seed of examples and incrementally adding small batches. Unfortunately, this can lead to data-sets that are biased to work well for one particular class of models and task, but may otherwise perform worse than a random set of examples (Settles, 2010, Section 6.6). Furthermore the active learning set-up can be prohibitively tedious and slow. To illustrate, Dwyer and Kondrak (2009) used 190 iterations of active learning to arrive at 2,000 words. Each iteration involves bootstrapping 10 different samples, and training 10 corresponding learners. Thus, in total, the underlying prediction model is trained 1,900 times. In contrast, our selection methods are fast, can select any number of data points in a single step, and are not tied to a particular prediction task or model. Furthermore, these methods can be combined with active learning in selecting the initial seed set.

Unsupervised Feature Selection Finally, we note that CSSP and related spectral methods have been applied to the problem of *unsupervised feature selection* (Stoppiglia et al., 2003; Mao, 2005; Wolf and Shashua, 2005; Zhao and Liu, 2007; Boutsidis et al., 2008). These methods are related to dimensionality reduction techniques such as Principal Components Analysis (PCA), but instead of truncating features in the eigenbasis representation (where each feature is a linear combination of all the original features), the goal is to remove dimensions in the standard basis, leading to a compact set of interpretable features. As long as the discarded features can be well approximated by a (linear) function of the selected features, the loss of information will be minimal.

Our first method for optimal data-set creation applies a randomized CSSP approach to the transpose of the data matrix, A^T. Equivalently, it selects the optimal k *rows* of A for embedding the full set of unlabeled examples. We use a recently developed randomized algorithm (Boutsidis et al., 2009), and an underlying rank-revealing QR factorization (Golub, 1965).

[2] We note that the discriminative model of Jiampojamarn and Kondrak (2010) outperforms the Bisani and Ney model by an average of about 0.75 percentage points across five data sets.

[3] `http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html`

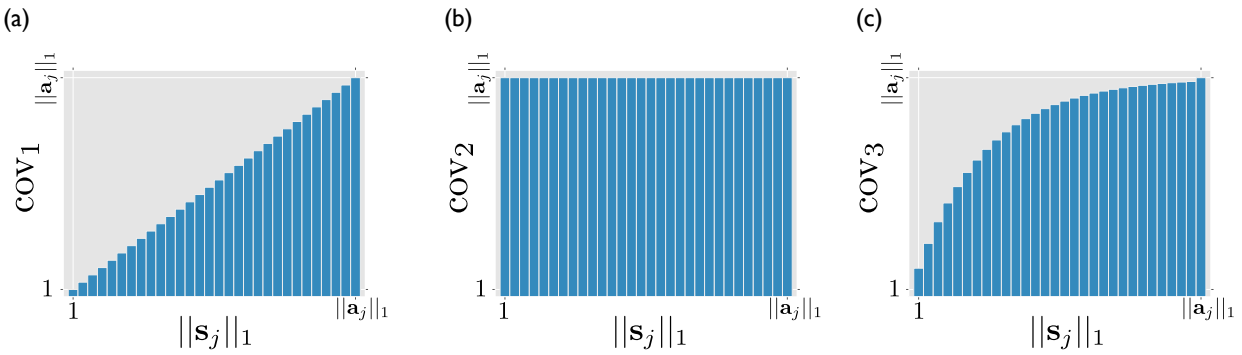

Figure 1: Various versions of the feature coverage function. Panel (a) shows cov_1 (Equation 5). Panel (b) shows cov_2 (Equation 6). Panel (c) shows cov_3 (Equation 7) with discount factor $\eta = 1.2$.

3 Two Methods for Optimal Data Set Selection

In this section we detail our two proposed methods for optimal data set selection. The key intuition is that we would like to pick a subset of data points which broadly and efficiently cover the features of the full range of data points. We assume a large pool $\mathcal{X}$ of n unlabeled examples, and our goal is to select a subset $\mathcal{S} \subset \mathcal{X}$ of size $k \ll n$ for labeling. We assume that each data point $x \in \mathcal{X}$ is a vector of m feature values. Our first method applies to any real or complex feature space, while our second method is specialized for binary features. We will use the $(n \times m)$ matrix A to denote our unlabeled data: each row is a data point and each column is a feature. In all our experiments, we used the presence (1) or absence (0) of each character 4-gram as our set of features.

3.1 Method 1: Row Subset Selection

To motivate this method, first consider the task of finding a rank k approximation to the data matrix A. The SVD decomposition yields:

$$A = U\Sigma V^T$$

- U is $(n \times n)$ orthogonal and its columns form the eigenvectors of AA^T
- V is $(m \times m)$ orthogonal and its columns form the eigenvectors of A^TA
- Σ is $(n \times m)$ diagonal, and its diagonal entries are the singular values of A (the square roots of the eigenvalues of both AA^T and A^TA).

To obtain a rank k approximation to A, we start by rewriting the SVD decomposition as a sum:

$$A = \sum_{i=1}^{\rho} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{1}$$

where $\rho = \min(m, n)$, σ_i is the i^{th} diagonal entry of Σ, $\mathbf{u}_i$ is the i^{th} column of U, and $\mathbf{v}_i$ is the i^{th} column of V. To obtain a rank k approximation to A, we simply truncate the sum in equation 1 to its first k terms, yielding A_k. To evaluate the quality of this approximation, we can measure the Frobenius norm of the residual matrix $||A - A_k||_F$.[4] The Eckart-Young theorem (Eckart and Young, 1936) states that A_k is optimal in the following sense:

$$A_k = \underset{\tilde{A} \text{ s.t. } \text{rank}(\tilde{A})=k}{\text{argmin}} ||A - \tilde{A}||_F \tag{2}$$

In other words, truncated SVD gives the best rank k approximation to A in terms of minimizing the Frobenius norm of the residual matrix. In CSSP, the goal is similar, with the added constraint that the approximation to A must be obtained by projecting onto the subspace spanned by a k-subset of the original rows of A.[5] Formally, the goal is to produce a $(k \times m)$ matrix S formed from rows of A, such that

$$||A - AS^+S||_F \tag{3}$$

[4] The Frobenius norm $||M||_F$ is defined as the entry-wise L_2 norm: $\sqrt{\sum_{i,j} m_{ij}^2}$

[5] Though usually framed in terms of column selection, we switch to row selection here as our goal is to select data points rather than features.

is minimized over all $\binom{n}{k}$ possible choices for S. Here S^+ is the $(m \times k)$ Moore-Penrose pseudo-inverse of S, and S^+S gives the orthogonal projector onto the rowspace of S. In other words, our goal is to select k data points which serve as a good approximate basis for *all* the data points. Since AS^+S can be at most rank k, the constraint considered here is stricter than that of Equation 1, so the truncated SVD A_k gives a lower bound on the residual.

Boutsidis et al (2009) develop a randomized algorithm that produces a submatrix S (consisting of k rows of A) which, with high probability, achieves a residual bound of:

$$||A - AS^+S||_F \leq O(k\sqrt{\log k})||A - A_k||_F \quad (4)$$

in running time $O(\min\{mn^2, m^2n\})$. The algorithm proceeds in three steps: first by computing the SVD of A, then by randomly sampling $O(k \log k)$ rows of A with importance weights carefully computed from the SVD, and then applying a deterministic rank-revealing QR factorization (Golub, 1965) to select k of the sampled rows. To give some intuition, we now provide some background on rank revealing factorizations.

Rank revealing QR / LQ (RRQR) Every real $(n \times m)$ matrix can be factored as $A = LQ$, where Q is $(m \times m)$ orthogonal and L is $(n \times m)$ lower triangular.[6] It is important to notice that in this triangular factorization, each successive row of A introduces exactly one new basis vector from Q. We can thus represent row i as a linear combination of the first $i - 1$ rows along with the ith row of Q.

A *rank-revealing* factorization is one which displays the numerical rank of the matrix — defined to be the singular value index r such that

$$\sigma_r \gg \sigma_{r+1} = O(\epsilon)$$

for machine precision ϵ. In the case of the LQ factorization, our goal is to order the rows of A such that each successive row has decreasing representational importance as a basis for the future rows. More formally, If there exists a row permutation Π such that ΠA has a triangular factorization $\Pi A = LQ$ with $L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}$, where the smallest singular value of L_{11} is much greater than the spectral norm of L_{22}, which is itself almost zero:

$$\sigma_{min}(L_{11}) \gg ||L_{22}||_2 = O(\epsilon)$$

then we say that $\Pi A = LQ$ is a *rank-revealing LQ factorization*. Both L_{11} and L_{22} will be lower triangular matrices and if L_{11} is $(r \times r)$ then A has numerical rank r (Hong and Pan, 1992).

[6] We replace the standard upper triangular QR factorization with an equivalent lower triangular factorization LQ to focus intuition on the rowspace of A.

Language	Training	Test	Total
Dutch	11,622	104,589	116,211
English	11209	100891	112100
French	2,748	24,721	27,469
Frisian	6,198	55,778	61,976
German	4,942	44,460	49,402
Italian	7,529	79,133	86,662
Norwegian	4,172	37,541	41,713
Spanish	3,150	28,341	31,491

Table 1: Pronunciation dictionary size for each of the languages.

Implementation In our implementation of the CSSP algorithm, we first prune away 4-gram features that appear in fewer than 3 words, then compute the SVD of the pruned data matrix using the PROPACK package,[7] which efficiently handles sparse matrixes. After sampling $k \log k$ words from A (with sampling weights calculated from the top-k singular vectors), we form a submatrix B consisting of the sampled words. We then use the RRQR implementation from ACM Algorithm 782 (Bischof and Quintana-Ortí, 1998) (routine DGEQPX) to compute $\Pi B = LQ$. We finally select the first k rows of ΠB as our optimal data set. Even for our largest data sets (English and Dutch), this entire procedure runs in less than an hour on a 3.4Ghz quad-core i7 desktop with 32 GB of RAM.

3.2 Method 2: Feature Coverage Maximization

In our previous approach, we adopted a general method for approximating a matrix with a subset of rows (or columns). Here we develop a novel objective function with the specific aim of optimal data set selection. Our key assumption is that the benefit of

[7] `http://soi.stanford.edu/~rmunk/PROPACK/`

seeing a new feature f in a selected data point bears a positive relationship to the frequency of f in the unlabeled pool of words. However, we further assume that the lion's share of benefit accrues quickly, with the marginal utility quickly tapering off as we label more and more examples with feature f. Note that for this method, we assume a boolean feature space.

To formalize this intuition, we will define the *coverage* of a selected $(k \times m)$ submatrix S consisting of rows of A, with respect to a feature index j. For illustration purposes, we will list three alternative definitions:

$$\text{cov}_1(S; j) = ||\mathbf{s}_j||_1 \tag{5}$$

$$\text{cov}_2(S; j) = ||\mathbf{a}_j||_1 \, \mathbb{I}\big(||\mathbf{s}_j||_1 > 0\big) \tag{6}$$

$$\text{cov}_3(S; j) = ||\mathbf{a}_j||_1 - \frac{||\mathbf{a}_j||_1}{\eta^{||\mathbf{s}_j||_1}} \, \mathbb{I}\big(||\mathbf{s}_j||_1 < ||\mathbf{a}_j||_1\big) \tag{7}$$

In all cases, $\mathbf{s}_j$ refers the j^{th} column of S, $\mathbf{a}_j$ refers the j^{th} column of A, $\mathbb{I}(\cdot)$ is a 0-1 indicator function, and η is a scalar discount factor.[8]

Figure 1 provides an intuitive explanation of these functions: cov_1 simply counts the number of selected data points with boolean feature j. Thus, full coverage ($||\mathbf{a}_j||$: the entire number of data points with the feature) is only achieved when *all* data points with the feature are selected. cov_2 lies at the opposite extreme. Even a single selected data point with feature j triggers coverage of the entire feature. Finally, cov_3 is designed so that the coverage scales monotonically as additional data points with feature j are selected. The first selected data point will capture all but $\frac{1}{\eta}$ of the total coverage, and each further selected data point will capture all but $\frac{1}{\eta}$ of whatever coverage remains. Essentially, the coverage for a feature scales as a geometric series in the number of selected examples having that feature.

To ensure that the total coverage ($||\mathbf{a}_j||_1$) is achieved when all the data points are selected, we add an indicator function for the case of $||\mathbf{c}_j||_1 = ||\mathbf{a}_j||_1$.[9]

[8]Chosen to be 5 in all our experiments. We experimented with several values between 2 and 10, without significant differences in results.

[9]Otherwise, the geometric coverage function would converge to $||\mathbf{a}_j||$ only as $||\mathbf{c}_j|| \to \infty$.

	500 Words			2000 Words		
	RAND	CSSP	FEAT	RAND	CSSP	FEAT
Dut	48.2	50.8	**59.3**	69.8	75.0	**77.8**
Eng	25.4	26.5	**29.5**	40.3	40.1	**42.8**
Fra	66.9	69.2	**72.1**	81.2	82.0	**84.8**
Fri	42.7	48.0	**53.6**	62.2	65.3	**68.5**
Ger	55.2	58.6	**65.0**	74.2	78.6	**80.8**
Ita	80.6	82.8	**82.8**	85.3	86.1	**86.8**
Nor	48.1	49.5	**55.0**	66.1	69.9	**71.6**
Spa	90.7	**96.8**	95.0	98.1	98.4	**99.0**
avg	57.2	60.3	**64.0**	72.2	74.4	**76.5**

Table 2: Test word accuracy across the 8 languages for randomly selected words (RAND), CSSP matrix subset selection (CSSP), and Feature Coverage Maximization (FEAT). We show results for 500 and 2000 word training sets.

Setting our feature coverage function to cov_3, we can now define the overall feature coverage of the selected points as:

$$\textbf{coverage}(S) = \frac{1}{||A||_1} \sum_j \text{cov}_3(S; j) \tag{8}$$

where $||A||_1$ is the L_1 entrywise matrix norm, $\sum_{i,j} |A_{ij}|$, which ensures that $0 \leq \textbf{coverage}(S) \leq 1$ with equality only achieved when $S = A$, i.e. when all data points have been selected.

We provide a brief sketch of our optimization algorithm: To pick the subset S of k words which optimizes Equation 8, we incrementally build optimal subsets $S' \subset S$ of size $k' < k$. At each stage, we keep track of the unclaimed coverage associated with each feature j:

$$\text{unclaimed}(j) = ||\mathbf{a}_j||_1 - \text{cov}_3(S'; j)$$

To add a new word, we scan through the pool of remaining words, and calculate the additional coverage that selecting word w would achieve:

$$\Delta(w) \quad = \sum_{\text{feature } j \text{ in } w} \text{unclaimed}(j) \left(\frac{\eta - 1}{\eta}\right)$$

We greedily select the word which adds the most coverage, remove it from the pool, and update the unclaimed feature coverages. It is easy to show that this greedy algorithm is globally optimal.

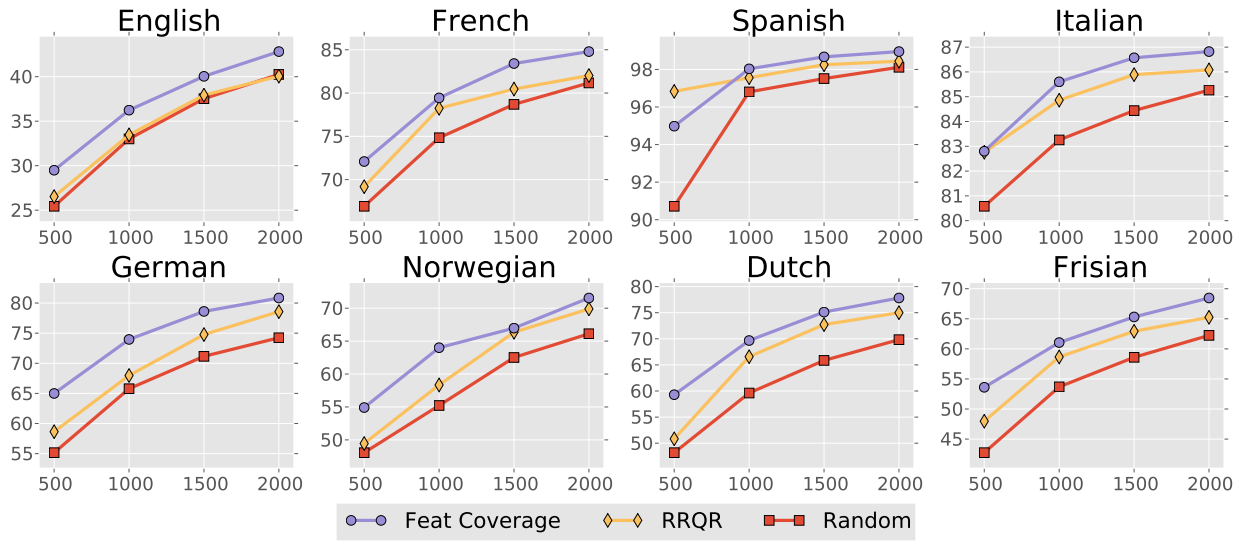

Figure 2: Test word accuracy across the 8 languages for (1) feature coverage, (2) CSSP matrix subset selection, (3) and randomly selected words.

4 Experiments and Analysis

To test the effectiveness of the two proposed data set selection methods, we conduct grapheme-to-phoneme prediction experiments across a test suite of 8 languages: Dutch, English, French, Frisian, German, Italian, Norwegian, and Spanish. The data was obtained from the PASCAL Letter-to-Phoneme Conversion Challenge,[10] and was processed to match the setup of Dwyer and Kondrak (2009). The data comes from a range of sources, including CELEX for Dutch and German (Baayen et al., 1995), BRULEX for French (Mousty et al., 1990), CMUDict for English,[11] the Italian Festival Dictionary (Cosi et al., 2000), as well as pronunciation dictionaries for Spanish, Norwegian, and Frisian (original provenance not clear).

As Table 1 shows, the size of the dictionaries ranges from 31,491 words (Spanish) up to 116,211 words (Dutch). We follow the PASCAL challenge training and test folds, treating the training set as our pool of words to be selected for labeling.

Results We consider training subsets of sizes 500, 1000, 1500, and 2000. For our baseline, we train the G2P model (Bisani and Ney, 2008) on randomly selected words of each size, and average the results over 10 runs. We follow the same procedure for our two data set selection methods. Figure 2 plots the word prediction accuracy for all three methods across the eight languages with varying training sizes, while Table 2 provides corresponding numerical results. We see that in all scenarios the two data set selection strategies fare better than random subsets of words.

In all but one case, the feature coverage method yields the best performance (with the exception of Spanish trained with 500 words, where the CSSP yields the best results). Feature coverage achieves average error reduction of 20% over the randomly selected training words across the different languages and training set sizes.

Coverage variants We also experimented with the other versions of the feature coverage function discussed in Section 3.2 (see Figure 1). While cov_1 tended to perform quite poorly (usually worse than random), cov_2 — which gives full credit for each feature the first time it is seen — yields results just slightly worse than the CSSP matrix method on average, and always better than random. In the 2000 word scenario, for example, cov_2 achieves average accuracy of 74.0, just a bit below the 74.4 accuracy of the CSSP method. It is also possible that more

[10] http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/

[11] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

	RAND	CSSP	FEAT	SVD
Fra	0.66	0.62	0.65	0.51
Fry	0.75	0.72	0.75	0.6
Ger	0.71	0.67	0.71	0.55
Ita	0.64	0.61	0.67	0.49
Nor	0.7	0.61	0.64	0.5
Spa	0.65	0.67	0.68	0.53
avg	0.69	0.65	0.68	0.53

Table 3: Residual matrix norm across 6 languages for randomly selected words (RAND), CSSP matrix subset selection (CSSP), feature coverage maximization (FEAT), and the rank k SVD (SVD). Lower is better.

	RAND	CSSP	FEAT
Dut	0.66	0.72	0.81
Eng	0.52	0.58	0.69
Fra	0.68	0.74	0.81
Fry	0.7	0.79	0.84
Ger	0.68	0.74	0.81
Ita	0.79	0.84	0.9
Nor	0.7	0.79	0.84
Spa	0.67	0.75	0.8
avg	0.68	0.74	0.81

Table 4: Feature coverage across the 8 languages for randomly selected words (RAND), CSSP matrix subset selection (CSSP), and feature coverage maximization (FEAT). Higher is better.

careful tuning of the discount factor η of cov_3 would yield further gains.

Optimization Analysis Both the CSSP and feature coverage methods have clearly defined objective functions — formulated in Equations 3 and 8, respectively. We can therefore ask how well each methods fares in optimizing either one of the two objectives.

First we consider the objective of the CSSP algorithm: to find k data points which can accurately embed the entire data matrix. Once the data points are selected, we compute the orthogonal projection of the data matrix onto the submatrix, obtaining an approximation matrix $\tilde{A}$. We can then measure the residual norm as a fraction of the original matrix norm:

$$\frac{||A - \tilde{A}||_F}{||A||_F} \tag{9}$$

As noted in Section 3.1, truncated SVD minimizes the residual over all rank k matrices, so we can compare our three methods — random selections, CSSP, and feature coverage — all of which select k examples as a basis, against the lower bound given by SVD. Table 3 shows the result of this analysis for $k = 2000$ (Note that we were unable to compute the projection matrices for English and Dutch due to the size of the data and memory limitations). As expected, SVD fares the best, with CSSP as a somewhat distant second. On average, feature coverage seems to do a bit better than random.

A similar analysis for the feature coverage objective function is shown in Table 4. Unsurprisingly, this objective is best optimized by the feature coverage method. Interestingly though, CSSP seems to perform about halfway between random and the feature coverage method. This makes some sense, as good basis data points will tend to have frequent features, while at the same time being maximally spread out from one another. We also note that the poor coverage result for English in Table 4 mirrors its overall poor performance in the G2P prediction task – not only are the phoneme labels unpredictable, but the input data itself is wild and hard to compress.

CSSP	FEAT	FEAT-SLS
fettered	internationalization	rating
exceptionally	underestimating	overs
gellert	schellinger	nation
daughtry	barristers	scherman
blowed	constellations	olinger
harmonium	complementing	anderson
cassini	bergerman	inter
rupees	characteristically	stated
tewksbury	heatherington	press
ley	overstated	conner

Table 5: Top 10 words selected by CSSP, feature coverage (FEAT), and feature coverage with stratified length sampling (FEAT-SLS)

Stratified length sampling As Table 5 shows, the top 10 words selected by the feature coverage method are mostly long and unusual, averaging 13.3 characters in length. In light of the potential annotation burden, we developed a stratified sampling strategy to ensure typical word lengths. Before selecting each new word, we first sample a word length according to the empirical word length distribution. We then choose among words of the sampled length

according to the feature coverage criterion. This results in more typical words of average length, with only a very small drop in performance.

5 Conclusion and Future Work

In this paper we proposed the task of optimal data set selection in the unsupervised setting. In contrast to active learning, our methods do not require repeated training of multiple models and iterative annotations. Since the methods are unsupervised, they also avoid tying the selected data set to a particular model class (or even task).

We proposed two methods for optimally selecting a small subset of examples for labeling. The first uses techniques developed by the numerical linear algebra and theory communities for approximating matrices with subsets of columns or rows. For our second method, we developed a novel notion of *feature coverage*. Experiments on the task of grapheme-to-phoneme prediction across eight languages show that our method yields performance improvements in all scenarios, averaging 20% reduction in error. For future work, we intend to apply the data set selection strategies to other NLP tasks, such as the optimal selection of sentences for tagging and parsing.

Acknowledgments

The authors thank the reviewers and acknowledge support by the NSF (grant IIS-1116676) and a research gift from Google. Any opinions, findings, or conclusions are those of the authors, and do not necessarily reflect the views of the NSF.

References

RH Baayen, R. Piepenbrock, and L. Gulikers. 1995. The celex lexical database (version release 2)[cd-rom]. *Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania*.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 5.

C.H. Bischof and G. Quintana-Ortí. 1998. Algorithm 782: codes for rank-revealing qr factorizations of dense matrices. *ACM Transactions on Mathematical Software (TOMS)*, 24(2):254–257.

C. Boutsidis, M.W. Mahoney, and P. Drineas. 2008. Unsupervised feature selection for principal components analysis. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–69.

C. Boutsidis, M. W. Mahoney, and P. Drineas. 2009. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. Society for Industrial and Applied Mathematics.

P. Cosi, R. Gretter, and F. Tesser. 2000. Festival parla italiano. *Proceedings of GFS2000, Giornate del Gruppo di Fonetica Sperimentale, Padova*.

K. Dwyer and G. Kondrak. 2009. Reducing the annotation effort for letter-to-phoneme conversion. In *Proceedings of the ACL*, pages 127–135. Association for Computational Linguistics.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of the Machine Translation Summit X*.

C. Eckart and G. Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

G. Golub. 1965. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7(3):206–216.

Yoo Pyo Hong and C-T Pan. 1992. Rank-revealing factorizations and the singular value decomposition. *Mathematics of Computation*, 58(197):213–232.

S. Jiampojamarn and G. Kondrak. 2010. Letter-phoneme alignment: An exploration. In *Proceedings of the ACL*, pages 780–788. Association for Computational Linguistics.

R.M. Kaplan and M. Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

J. Kominek and A. W. Black. 2006. Learning pronunciation dictionaries: language complexity and word selection strategies. In *Proceedings of the NAACL*, pages 232–239. Association for Computational Linguistics.

K.Z. Mao. 2005. Identifying critical variables of principal components for unsupervised feature selection. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(2):339–344.

Y. Marchand and R.I. Damper. 2000. A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26(2):195–219.

P. Mousty, M. Radeau, et al. 1990. Brulex. une base de données lexicales informatisée pour le français écrit et parlé. *L'année psychologique*, 90(4):551–566.

T.J. Sejnowski and C.R. Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168.

Burr Settles. 2010. Active learning literature survey. Technical Report TR1648, Department of Computer Sciences, University of Wisconsin-Madison.

H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar. 2003. Ranking a random feature for variable and feature selection. *The Journal of Machine Learning Research*, 3:1399–1414.

L. Wolf and A. Shashua. 2005. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *The Journal of Machine Learning Research*, 6:1855–1887.

Z. Zhao and H. Liu. 2007. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the ICML*, pages 1151–1157.