Compact Lexicon Selection with Spectral Methods

Young-Bum Kim[†] **Karl Stratos**[‡] **Xiaohu Liu**[†] **Ruhi Sarikaya**[†]

[†]Microsoft Corporation, Redmond, WA
[‡]Columbia University, New York, NY
{ybkim, derekliu, ruhi.sarikaya}@microsoft.com
stratos@cs.columbia.edu

Abstract

In this paper, we introduce the task of selecting compact lexicon from large, noisy gazetteers. This scenario arises often in practice, in particular spoken language understanding (SLU). We propose a simple and effective solution based on matrix decomposition techniques: canonical correlation analysis (CCA) and rank-revealing QR (RRQR) factorization. CCA is first used to derive low-dimensional gazetteer embeddings from domain-specific search logs. Then RRQR is used to find a subset of these embeddings whose span approximates the entire lexicon space. Experiments on slot tagging show that our method yields a small set of lexicon entities with average relative error reduction of $> 50\%$ over randomly selected lexicon.

1 Introduction

Discriminative models trained with large quantities of arbitrary features are a dominant paradigm in spoken language understanding (SLU) (Li et al., 2009; Hillard et al., 2011; Celikyilmaz et al., 2013; Liu and Sarikaya, 2014; Sarikaya et al., 2014; Anastasakos et al., 2014; Xu and Sarikaya, 2014; Celikyilmaz et al., 2015; Kim et al., 2015a; Kim et al., 2015c; Kim et al., 2015b). An important category of these features comes from *entity dictionaries* or *gazetteers*—lists of phrases whose labels are given. For instance, they can be lists of movies, music titles, actors, restaurants, and cities. These features enable SLU models to robustly handle unseen entities at test time.

However, these lists are often massive and very noisy. This is because they are typically obtained automatically by mining the web for recent entries (such as newly launched movie names). Ideally, we would like an SLU model to have access to this vast source of information at deployment. But this is difficult in practice because an SLU model needs to be light-weight to support fast user interaction. It becomes more challenging when we consider multiple domains, languages, and locales.

In this paper, we introduce the task of selecting a small, representative subset of noisy gazetteers that will nevertheless improve model performance nearly as much as the original lexicon. This will allow an SLU model to take full advantage of gazetteer resources at test time without being overwhelmed by their scale.

Our selection method is two steps. First, we gather relevant information for each gazetteer element using domain-specific search logs. Then we perform CCA using this information to derive low-dimensional gazetteer embeddings (Hotelling, 1936). Second, we use a subset selection method based on RRQR to locate gazetteer embeddings whose span approximates the the entire lexicon space (Boutsidis et al., 2009; Kim and Snyder, 2013). We show in slot tagging experiments that the gazetteer elements selected by our method not only preserve the performance of using full lexicon but even improve it in some cases. Compared to random selection, our method achieves average relative error reduction of $> 50\%$.

2 Motivation

We motivate our task by describing the process of lexicon construction. Entity dictionaries are usually automatically mined from the web using resources that provide typed entities. On a regular basis, these dictionaries are automatically updated and accumulated based on local data feeds and knowledge graphs. Local data feeds are generated from various origins (e.g., yellow pages, Yelp). Knowledge graphs such as `www.freebase.com` are resources that define a semantic space of entities (e.g., movie names, per-

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 806–811,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

sons, places and organizations) and their relations.

Because of the need to keep dictionaries updated to handle newly emerging entities, lexicon construction is designed to aim for high recall at the expense of precision. Consequently, the resulting gazetteers are noisy. For example, a movie dictionary may contain hundreds of thousands movie names, but many of them are false positives.

While this large base of entities is useful as a whole, it is challenging to take advantage of at test time. This is because we normally cannot afford to consume so much memory when we deploy an SLU model in practice. In the next section, we will describe a way to filter these entities while retaining their overall benefit.

3 Method

3.1 Row subset selection problem

We frame gazetteer element selection as the row subset selection problem. In this framework, we organize n gazetteer elements as matrix $A \in \mathbb{R}^{n \times d}$ whose rows $A_i \in \mathbb{R}^d$ are some representations of the gazetteer members. Given $m \leq n$, let $\mathcal{S}(A, m) := \{B \in \mathbb{R}^{m \times d} : B_i = A_{\pi(i)}\}$ be a set of matrices whose rows are a subset of the rows of A. Note that $|\mathcal{S}(A, m)| = \binom{n}{m}$. Our goal is to select [1]

$$B^* = \underset{B \in \mathcal{S}(A,m)}{\arg\min} \left|\left|A - AB^+B\right|\right|_F$$

That is, we want B to satisfy range$(B^\top) \approx$ range$(A^\top)$. We can solve for B^* exactly with exhaustive search in $O(n^m)$, but this brute-force approach is clearly not scalable. Instead, we turn to the $O(nd^2)$ algorithm of Boutsidis et al. (2009) which we review below.

3.1.1 RRQR factorization

A key ingredient in the algorithm of Boutsidis et al. (2009) is the use of RRQR factorization. Recall that a (thin) QR factorization of A expresses $A = QR$ where $Q \in \mathbb{R}^{n \times d}$ has orthonormal columns and $R \in \mathbb{R}^{d \times d}$ is an upper triangular matrix. A limitation of QR factorization is that it does not assign a score to each of the d components. This is in contrast to singular value decomposition (SVD) which assigns a score (singular value) indicating the importance of these components.

Input: d-dimensional gazetteer representations $A \in \mathbb{R}^{n \times d}$, number of gazetteer elements to select $m \leq n$
Output: m rows of A, call $B \in \mathbb{R}^{m \times d}$, such that $\left|\left|A - AB^+B\right|\right|_F$ is small

- Perform SVD on A and let $U \in \mathbb{R}^{d \times m}$ be a matrix whose columns are the left singular vectors corresponding to the largest m singular values.
- Associate a probability p_i with the i-th row of A as follows:
$$p_i := \min\left\{1, \lfloor m \log m \rfloor \frac{||U_i||^2}{m}\right\}$$
- Discard the i-th row of A with probability $1 - p_i$. If kept, the row is multiplied by $1/\sqrt{p_i}$. Let these $O(m \log m)$ rows form the columns of a new matrix $\bar{A} \in \mathbb{R}^{d \times O(m \log m)}$.
- Perform RRQR on $\bar{A}$ to obtain $\bar{A}\Pi = QR$.
- Return the m rows of the original A corresponding to the top m columns of $\bar{A}\Pi$.

Figure 1: Gazetteer selection based on the algorithm of Boutsidis et al. (2009).

RRQR factorization is a less well-known variant of QR that addresses this limitation. Let $\sigma_i(M)$ denote the i-th largest singular value of matrix M. Given A, RRQR jointly finds a permutation matrix $\Pi \in \{0, 1\}^{d \times d}$, orthonormal $Q \in \mathbb{R}^{n \times d}$, and upper triangular $R = [R_{11}\ R_{12};\ 0\ R_{22}] \in \mathbb{R}^{d \times d}$ such that

$$A\Pi = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$$

satisfying $\sigma_k(R_{11}) = O(\sigma_k(A))$ and $\sigma_1(R_{22}) = \Omega(\sigma_{k+1}(A))$ for $k = 1 \dots d$. Because of this ranking property, RRQR "reveals" the numerical rank of A. Furthermore, the columns of $A\Pi$ are sorted in the order of decreasing importance.

3.1.2 Gazetteer selection algorithm

The algorithm is a two-stage procedure. In the first step, we randomly sample $O(m \log m)$ rows of A with carefully chosen probabilities and scale them to form columns of matrix $\bar{A} \in \mathbb{R}^{d \times O(m \log m)}$. In the second step, we perform RRQR factorization on $\bar{A}$ and collect the gazetteer elements corresponding to the top components given by the RRQR permutation. The algorithm is shown in Figure 1. The first stage involves random sampling and scaling of rows, but it is shown that $\bar{A}$

[1] The Frobenius norm $||M||_F$ is defined as the entry-wise L_2 norm: $\sqrt{\sum_{i,j} m_{ij}^2}$. B^+ is the Moore-Penrose pseudo-inverse of B

has $O(m \log m)$ columns with constant probability.

This algorithm has the following optimality guarantee:

Theorem 3.1 (Boutsidis et al. (2009)). *Let $\hat{B} \in \mathbb{R}^{m \times d}$ be the matrix returned by the algorithm in Figure 1. Then with probability at least 0.7,*

$$\left|\left|A - A\hat{B}^{+}\hat{B}\right|\right|_F \leq O(m\sqrt{\log m}) \times \min_{\substack{\tilde{A} \in \mathbb{R}^{n \times d}: \\ rank(\tilde{A})=m}} \left|\left|A - \tilde{A}\right|\right|_F$$

In other words, the selected rows are not arbitrarily worse than the best rank-m approximation of A (given by SVD) with high probability.

3.2 Gazetteer embeddings via CCA

In order to perform the selection algorithm in Figure 1, we need a d-dimensional representation for each of n gazetteer elements. We use CCA for its simplicity and generality.

3.2.1 Canonical Correlation Analysis (CCA)

CCA is a general statistical technique that characterizes the linear relationship between a pair of multi-dimensional variables. CCA seeks to find k dimensions (k is a parameter to be specified) in which these variables are maximally correlated.

Let $x_1 \dots x_n \in \mathbb{R}^d$ and $y_1 \dots y_n \in \mathbb{R}^{d'}$ be n samples of the two variables. For simplicity, assume that these variables have zero mean. Then CCA computes the following for $i = 1 \dots k$:

$$\underset{\substack{u_i \in \mathbb{R}^d,\, v_i \in \mathbb{R}^{d'}: \\ u_i^\top u_{i'}=0 \;\; \forall i'<i \\ v_i^\top v_{i'}=0 \;\; \forall i'<i}}{\arg\max} \frac{\sum_{l=1}^{n} (u_i^\top x_l)(v_i^\top y_l)}{\sqrt{\sum_{l=1}^{n} (u_i^\top x_l)^2}\sqrt{\sum_{l=1}^{n} (v_i^\top y_l)^2}}$$

In other words, each (u_i, v_i) is a pair of projection vectors such that the *correlation* between the projected variables $u_i^\top x_l$ and $v_i^\top y_l$ is maximized, under the constraint that this projection is *uncorrelated* with the previous $i - 1$ projections.

This is a non-convex problem due to the interaction between u_i and v_i. However, a method based on singular value decomposition (SVD) provides an efficient and exact solution to this problem (Hotelling, 1936). The resulting solution $u_1 \dots u_k \in \mathbb{R}^d$ and $v_1 \dots v_k \in \mathbb{R}^{d'}$ can be used to project the variables from the original d- and d'-dimensional spaces to a k-dimensional space:

$$x \in \mathbb{R}^d \longrightarrow \bar{x} \in \mathbb{R}^k : \bar{x}_i = u_i^\top x$$
$$y \in \mathbb{R}^{d'} \longrightarrow \bar{y} \in \mathbb{R}^k : \bar{y}_i = v_i^\top y$$

The new k-dimensional representation of each variable now contains information about the other variable. The value of k is usually selected to be much smaller than d or d', so the representation is typically also low-dimensional.

3.2.2 Inducing gazetteer embeddings

We now describe how to use CCA to induce vector representations for gazetteer elements. Using the same notation, let n be the number of elements in the entire gazetteers. Let $x_1 \dots x_n$ be the original representations of the element samples and $y_1 \dots y_n$ be the original representations of the associated features in the element.

We employ the following definition for the original representations. Let d be the number of distinct element types and d' be the number of distinct feature types.

- $x_l \in \mathbb{R}^d$ is a zero vector in which the entry corresponding to the element type of the l-th instance is set to 1.
- $y_l \in \mathbb{R}^{d'}$ is a zero vector in which the entries corresponding to features generated by the element are set to 1.

In our case, we want to induce gazetteer (element) embeddings that correlate with the relevant features about gazetteers. For this purpose, we use three types of features: context features, search click log features, and knowledge graph features.

Context features: For each gazetteer element g of domain l, we take sentences from search logs on domain l containing g and extract five words each to the left and the right of the element g in the sentences. For instance, if $g =$ “The Matrix” is a gazetteer element of domain $l =$ “Movie”, we collect sentences from movie-specific search logs involving the phrase “The Matrix”. Such domain-specific search logs are collected using a pre-trained domain classifier.

Search click log features: Large-scale search engines such as Bing and Google process millions of queries on a daily basis. Together with the search queries, user clicked URLs are also logged anonymously. These click logs have been used for extracting semantic information for various NLP tasks (Kim et al., 2015a; Tseng et al., 2009; Hakkani-Tür et al., 2011). We used the clicked URLs as features to determine the likelihood of an entity being a member of a dictionary. These features are useful because common URLs are shared across different names such as movie, business and music. Table 1 shows the top five most frequently clicked URLs for movies "Furious 7" and "The age of adaline".

Furious 7	**The age of adaline**
imdb.com	imdb.com
en.wikipedia.org	en.wikipedia.org
furious7.com	youtube.com
rottentomatoes.com	rottentomatoes.com
www.msn.com	movieinsider.com

Table 1: Top clicked URLs of two movies.

One issue with using only click logs is that some entities may not be covered in the query logs since logs are extracted from a limited time frame (e.g. six months). Even the big search engines employ a moving time window for processing and storing search logs. Consequently, click logs are not necessarily good evidence. For example, "apollo thirteen" is a movie name appearing in the movie training data, but it does not appear in search logs. One way to solve the issue of missing logs for entities is to search `bing.com` at real time. Given that the search engine is updated on a daily basis, real-time search can make sure we capture the newest entities. We run live search for all entities no matter if they appear in search logs or not. Each URL returned from the live search is considered to have an additional click.

Knowledge graph features: The graph in `www.freebase.com` contains a large set of tuples in a resource description framework (RDF) defined by W3C. A tuple typically consists of two entities: a subject and an object linked by some relation.

An interesting part of this resource is the entity type defined in the graph for each entity. In the knowledge graph, the "type" relation represents the entity type. Table 2 shows some examples of entities and their relations in the knowledge graph. From the graph, we learn that "Romeo & Juliet" could be a film name or a music album since it has two types: "film.film" and "music.album".

Subject	**Relation**	**Object**
Jason Statham	type	film.actor
Jason Statham	type	tv.actor
Jason Statham	type	film.producer
Romeo & Juliet	type	film.film
Romeo & Juliet	type	music.album

Table 2: Entities & relation in the knowledge graph.

4 Experiments

To test the effectiveness of the proposed gazetteer selection method, we conduct slot tagging experiments across a test suite of three domains: Movies, Music and Places, which are very sensitive domains to gazetteer features. The task of slot tagging is to find the correct sequence of tags of words given a user utterance. For example, in Places domain, a user could say "search for home depot in kingsport" and the phrase "home depot" and "kingsport" are tagged with `Place_Name` and `Location` respectively. The data statistics are shown in Table 3. One domain can have various kinds of gazetteers. For example, Places domain has business name, restaurant name, school name and etc. Candidate dictionaries are mined from the web and search logs automatically using basic pattern matching approaches (e.g. entities sharing the same or similar context in queries or documents) and consequently contain significant amount of noise. As the table indicates, the number of elements in total across all the gazetteers (#total gazet elements) in each domain are too large for models to consume.

In all our experiments, we trained conditional random fields (CRFs) (Lafferty et al., 2001) with the following features: (1) n-gram features up to $n = 3$, (2) regular expression features, and (3) Brown clusters (Brown et al., 1992) induced from search logs. With these features, we compare the following methods to demonstrate the importance of adding appropriate gazetteers:

- *NoG*: train without gazetteer features.
- *AllG*: train with all gazetteers.
- *RandG*: train with randomly selected gazetteers.
- *RRQRG*: train with gazetteers selected from RRQR.
- *RankAllG*: train with all ranked gazetteers.

Domains	#labels	#kinds of gazets	#total gazet elements	#training queries	#test queries
Movies	25	21	14,188,527	43,784	12,179
Music	7	13	62,231,869	31,853	8,615
Places	32	31	34,227,612	22,345	6,143

Table 3: Data statistics

Here gazetteer features are activated when a phrase contains an entity in a dictionary. For RandG, we first sample a category of gazetteers uniformly and then choose a lexicon from gazetteers in that category. The results when we use selected gazetteer randomly in whole categories are very low and did not include them here. For selecting gazetteer methods (NoG, RnadG and RRQRG), we select 500,000 elements in total.

	Places	Music	Movies	AVG.
NoG	89.10	81.53	84.78	85.14
AllG	92.11	84.24	88.56	88.30
RRQRG	91.80	83.83	87.41	87.68
RandG	86.20	76.53	77.23	79.99

Table 4: Comparison of models evaluated on three domains. The numbers are F1-scores.

4.1 Results across Domains

First, we evaluate all models across three domains. Note that the both training and test data are collected from the United States. The results are shown in Table 4. Not surprisingly, using all gazetteer features (AllG) boosts the F1 score from 85.14 % to 88.30%, confirming the power of gazetteer features. However, with a random selection of gazetteers, the model does not perform well, only achieving 79.99% F1-score. Interestingly, we see that across all domains our method (RRQRG) fares better than both RandG and NoG, almost reaching the AllG performance with gazetteer size dramatically reduced.

4.2 Results across Locales

In the next experiments, we run experiments across three different locales in Places domain: United Kingdom (GB), Australia (AU), and India (IN). The Places is a very sensitive domain to locales[2]. For example, restaurant names in India are very different from Australia. Here we assume that unlike the previous experiments, the training data is collected from the United States and test data is collected from different locales. We used same training data in the previous experiments and the size of test data is about 5k for each locale. The results are shown in Table 5. Interestingly, the RRQR even outperforms the AllG. This is because some noisy entities are filtered.

Finally, we show that the proposed method is useful even in all gazetteer scenario (AllG). Using RRQR, we can order entities according to their importance and transform a gazetteer feature into a few ones by binning the entities with their rankings. For example, instead of having one single big business names gazetteer, we can divide them into lexicon with first 1000 entities, 10000 entities and so on. Results using ranked gazetteers are shown in Table 6. We see that the Ranked gazetteers approach (RankAllG) has consistent gains across domains over AllG.

	GB	AU	IN
NoG	87.70	82.20	80.30
AllG	90.12	86.98	89.77
RRQRG	90.18	87.48	90.28
RandG	86.20	65.34	64.20

Table 5: Comparison of models across different locales.

	Places	Music	Movies	AVG.
AllG	92.11	84.24	88.56	88.30
RankAllG	92.78	86.30	89.1	89.40

Table 6: Comparison of models with or without ranked gazetteers. These are evaluated on three domains collected in the United States.

5 Conclusion

We proposed the task of selecting compact lexicons from large and noisy gazetteers. This scenario arises often in practice. We introduced a simple and effective solution based on matrix decomposition techniques: CCA is used to derive low-dimensional gazetteer embeddings and RRQR is used to find a subset of these embeddings. Experiments on slot tagging show that our method yields relative error reduction of $> 50\%$ on average over the random selection method.

[2] Since it is very difficult to create all locale specific training data, gazetteer features are very crucial.

References

Tasos Anastasakos, Young-Bum Kim, and Anoop Deoras. 2014. Task specific continuous word representations for mono and multi-lingual spoken language understanding. In *ICASSP*, pages 3246–3250. IEEE.

Christos Boutsidis, Michael W Mahoney, and Petros 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.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Asli Celikyilmaz, Dilek Z Hakkani-Tür, Gökhan Tür, and Ruhi Sarikaya. 2013. Semi-supervised semantic tagging of conversational understanding using markov topic regression. In *ACL (1)*, pages 914–923.

Asli Celikyilmaz, Dilek Hakkani-Tur, Panupong Pasupat, and Ruhi Sarikaya. 2015. Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems. AAAI - Association for the Advancement of Artificial Intelligence, January.

Dilek Hakkani-Tür, Gokhan Tur, Larry Heck, Asli Celikyilmaz, Ashley Fidler, Dustin Hillard, Rukmini Iyer, and S. Parthasarathy. 2011. Employing web search query click logs for multi-domain spoken language understanding. IEEE Automatic Speech Recognition and Understanding Workshop, December.

Dustin Hillard, Asli Celikyilmaz, Dilek Z Hakkani-Tür, and Gökhan Tür. 2011. Learning weighted entity lists from web click logs for spoken language understanding. In *INTERSPEECH*, pages 705–708.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Young-Bum Kim and Benjamin Snyder. 2013. Optimal data set selection: An application to grapheme-to-phoneme conversion. In *HLT-NAACL*, pages 1196–1205. Association for Computational Linguistics.

Young-Bum Kim, Jeong Minwoo, Karl Startos, and Ruhi Sarikaya. 2015a. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *HLT-NAACL*, pages 84–92. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2015b. Pre-training of hidden-unit crfs. In *ACL*. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015c. New transfer learning techniques for disparate label sets. In *ACL*. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Xiaohu Liu and Ruhi Sarikaya. 2014. A discriminative model based entity dictionary weighting approach for spoken language understanding. In *Spoken Language Technology Workshop (SLT)*, pages 195–199. IEEE.

Ruhi Sarikaya, Asli C, Anoop Deoras, and Minwoo Jeong. 2014. Shrinkage based features for slot tagging with conditional random fields. In Proceeding of ISCA - International Speech Communication Association, September.

Huihsin Tseng, Longbin Chen, Fan Li, Ziming Zhuang, Lei Duan, and Belle Tseng. 2009. Mining search engine clickthrough log for matching n-gram features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 524–533. Association for Computational Linguistics.

Puyang Xu and Ruhi Sarikaya. 2014. Targeted feature dropout for robust slot filling in natural language understanding. In *ISCA - International Speech Communication Association*, September.