# SPARSE LEXICAL REPRESENTATION FOR SEMANTIC ENTITY RESOLUTION

*Yuzhe Jin, Kuansan Wang, and Emre Kıcıman*

Microsoft Research, Redmond, USA
{yuzjin,kuansanw,emrek}@microsoft.com

**ABSTRACT**

This paper addresses the problem of semantic entity resolution (SER), which aims to determine whether some or none of the entities in a knowledge base is mentioned in a given web document. The lexical features, e.g., words and phrases, which are critical to the resolution of the semantic entities are typically of a small amount compared to all lexical features in the web document, and therefore can be modeled as sparse signals. Two techniques leveraging the principles of sparse signal recovery are proposed to identify the sparse, salient lexical features: one technique, based on the Lasso algorithm with the $\ell_2$-norm distance metric, attempts to recover all the salient lexical features at once; the other technique, namely Posterior Probability Pursuit (PPP), sequentially identifies salient features one after one using the negative log posterior probability as the distance metric. Using a knowledge base consisting of about 100 million entities, we show that the proposed techniques exploiting the sparsity nature underlying SER deliver substantial performance improvement over baseline methods without sparsity consideration, demonstrating the potentials of sparse signal techniques in entity-centric web information processing.

***Index Terms*—** Sparse signal recovery, semantic entity resolution, posterior probability pursuit, Lasso

## 1. INTRODUCTION

The area of sparse signal recovery has received much research attention in the past several decades [1–4]. The underlying problem is to recover a sparse signal, whose vector representation has only a small number of nonzero entries, based on as few measurements as possible. This problem has driven the advancements of a wide spectrum of applications such as compressed sensing [5, 6], medical imaging [7], face recognition [8], robust regression [9], speech coding [10], body area networks [11], echo cancellation [12], and wireless communication [13, 14].

The vastness and variety of the information available on the web has recently provided an unprecedented opportunity for the theories and algorithms of sparse signal recovery in advancing web-scale information processing applications. Different from traditional applications of sparse signal techniques in which the signals of interest are usually real or complex vectors, web information possesses various forms of signals, with texts being the most common. Accordingly, tackling web information requires proper transformation of the discrete signals such as texts into representations that sparse signal techniques can handle, and typical techniques for parameterizing text documents into vector representations [15] include the term-document matrix [16–18] as well as the language model [19]. Kasiviswanathan et al. [16] applied the technique for sparse signal recovery to detect emerging topic in streaming user-generated contents. If a document can find a sparse representation using a large collection of previous documents, the topic of the document is not novel. If, however, a sparse representation cannot be obtained, the document is very likely to be focusing on a newly emerging topic that has not been covered by the pool of existing documents. In [17], Min et al. utilizes a low-rank and sparse matrix decomposition technique to decompose the background topics from keywords on a set of documents. The topical background shared across multiple documents is modeled by a low-rank matrix, whereas the keywords specifically related to each individual document is captured by a sparse matrix, where the sparsity results from the small amount of keywords compared with abundant background information. Agarwal and Gurevich [18] applied the idea of sparsification to re-parameterize a set of documents with sparser vector-space representation so that the top-$k$ retrieval can be efficiently implemented in the proposed recommendation system. Hutchinson et al. [19] proposed a novel maximum entropy language model that decomposes the model parameters into a low rank component that learns regularities in the training data and a sparse component that learns exceptions, and showed that the proposed model effectively reduces the perplexity. With the application of face detection, Le et al. [20] proposed a large scale unsupervised learning approach to building high level, class-specific feature detectors, where the sparsity constraint is employed to effectively ensure the desired model structure.

This paper studies the problem of semantic entity resolution (SER), whose goal is to determine whether some or none of entities in a knowledge base is mentioned in a given web document. Successful solutions to the SER problem build an important step to enabling novel user experiences in web information retrieval and data mining. Previous research [21–25] studied the problem of resolving entity mentions against knowledge bases. The main difficulty originates from the fact that an entity can have multiple surface forms while different entities can share a same surface form. This many-to-many relationship leads to an inherent semantic ambiguity. To address this problem, for instance, Mihalcea and Csomai [26] considered a knowledge-based approach which computes the contextual overlap between the entity definitions and the words to disambiguate, and also an approach that adopts features including part-of-speech, local contexts with specified locations, among others. Cucerzan [27] proposed an approach to solve the named entity disambiguation problem using the Wikipedia knowledge base. Using a vector space model, the system maximizes the agreement between each of the mentions and the document in terms of contextual information as well as all pairwise agreements between mentions in terms of category information. Han and Sun [28] employs a generative entity-mention model for linking entity mentions to a knowledge base, which can incorporate the prior popularity of entities, name variations, and knowledge base entries. A Naive Bayes classifier is used with the generative model to determine the best matching entity in the knowledge base.

Note that existing approaches for entity resolution use either all the lexical features in a web document [29] or the lexical features within a window from the entity mentions [26] to resolve the semantics. Empirically, however, only a few lexical features in a document turn out to be most critical to the resolution of entity mentions, whereas the rest large amount of lexical features are language components either less important or completely unrelated to resolv-

ing entity mentions. The small amount of salient lexical features critical to SER can be modeled as sparse signals. We propose two different approaches, both rooted in sparse signal recovery, to identify and thus directly utilize the sparse salient features for SER. The first approach is based on the Lasso algorithm, which attempts to jointly determine all sparse salient features at once. The second approach, namely Posterior Probability Pursuit (PPP), sequentially identifies the salient features one after one. More importantly, PPP employs the negative log posterior probability as the distance metric, in contrast to the popular $\ell_2$-norm widely used in sparse signal recovery. We perform an experiment using an entity knowledge consisting of about 100 million people entities. The preliminary results demonstrate that the proposed sparse signal recovery techniques have strong potentials to improve SER performance. Further, the posterior probability is a more suitable metric than the $\ell_2$-norm in the entity-centric web information processing task.

## 2. SEMANTIC ENTITY RESOLUTION

### 2.1. Problem Formulation

Let $\mathcal{E}$ denote an entity knowledge base. For each entity $e \in \mathcal{E}$, the information is represented in the form of attribute-value pairs. Let $a_e$ denote an attribute of $e$, and $v_e$ denote the corresponding value of attribute $a_e$. For example, for a people entity $e$, $a_e$ may be the attribute "gender", and $v_e$ can be "male" as the value of the attribute. In this paper, we assume all values are text-based, although the values can take on other forms such as image, audio, or video. Using this representation, an entity $e$ in the knowledge base can be characterized by $\{(a_e^{(k)}, v_e^{(k)})\}_{k=1}^{r}$, where $r$ is the number of attribute-value pairs available for entity $e$. For a given web document $D$, the goal is to determine an entity $e \in \mathcal{E}$ so that $D$ mentions $e$ in its content. If no such entity exists, we claim that $D$ mentions an entity that is outside of $\mathcal{E}$, or simply an *unknown* entity.

It is common to apply proper heuristics to confine the search space to a set of entities that are most likely to contain the corresponding entity. For instance, an inverted index can be exploited to retrieve a set of entities which have at least certain feature overlap with the document [15]. Suppose we can obtain such a confined entity set, denoted by $\mathcal{E}_c$ with $\mathcal{E}_c \subseteq \mathcal{E}$, which is typically a much smaller subset of the original entity knowledge base $\mathcal{E}$.

### 2.2. Lexical Features

We utilize lexical features to perform semantic entity resolution. All the (text-based) values (i.e., $v_e$'s) in the knowledge base serve as the source of potential lexical features. We adopt a bag of words and phrases model, in which a lexical feature can be either a word or a phrase, where the latter can naturally preserve the dependency across words. To build such a model, we need to systematically harvest phrases from the text-based values of an entity. Two approaches for phrase discovery are adopted. The first approach relies on the observation that some attribute $a_e$ usually takes a phrase as its value $v_e$. For example, in the LinkedIn knowledge base, attributes such as "university attended", "degree earned", and "job title" typically have values which are by themselves phrases, e.g., "University of California, San Francisco", "Master of Science", and "senior hardware engineer", respectively. In this case, the value $v_e$ as a whole can be treated as a reasonable phrase. An attribute-value pair in the Wikipedia Infobox is another example of this type of source of phrases. The second approach for phrase discovery requires considerable effort since it extracts phrases from values with free-style texts, such as the main text body of a Wikipedia entry, a detailed review of an restaurant on Yelp, and a summary of a persons's professional experiences on LinkedIn. The phrases are extracted through the application of a statistical language model, namely a phrase LM, which models phrase boundaries as partially observable variables [30]. A phrase segmentation is a set of contiguous phrases which together constitute a particular free-style text. Using a phrase LM, optimal phrase segmentations can be found through dynamic programming. The details are omitted due to the scope of the paper.

### 2.3. Sparsity Nature in Entity Resolution

The fundamental assumption which motivates this work is that the salient lexical features that are critical to the resolution of the entity mention is of a small amount, and hence sparse, compared with all lexical features of the web document. As an example, let us examine the following text.

"*By acclamation, Michael Jordan is the greatest basketball player of all time. Although, a summary of his basketball career and influence on the game inevitably fails to do it justice, as a phenomenal athlete with a unique combination of fundamental soundness, grace, speed, power, artistry, improvisational ability and an unquenchable competitive desire, Jordan single-handedly redefined the NBA superstar.*"

A reader would have little difficulty in recognizing that the *Michael Jordan* mentioned above is the well-known NBA basketball player. The salient lexical features leading to the resolution are most likely "Michael Jordan", "basketball", "NBA", and "superstar". The rest of the lexical components are either language components which are merely related to the entity (e.g., "a summary of", "a unique combination of"), or features that are less critical (and more ambiguous) once the most salient features are identified (e.g., "speed", "power", "artistry", "inevitably fails to do it justice"). The salient lexical features that can uniquely determine the entity are actually surrounded by the lexical components that are less useful for disambiguating the semantics, and are therefore sparse. Meanwhile, although the knowledge base can provide a comprehensive coverage of the information pertaining to an entity, a web document may only mention some, usually not all, of the attributes of the entity. In this example, lexical features such as "Chicago Bulls" in Michael Jordan's knowledge entry (e.g., Wikipedia) do not appear. Therefore, rather than considering all lexical components available in the text, we should focus on the sparse salient lexical features whose joint presence suffices to uniquely resolve the entity ambiguity. The fertile area of sparse signal recovery offers useful tools for taking advantage of the sparsity nature underlying semantic entity resolution. Next, we explore two approaches for SER, both of which are motivated by popular sparse signal recovery techniques.

### 2.4. Lasso-Based Semantic Entity Resolution

We begin with a preparatory setup. Denote by $\mathcal{H}_e$ the set of phrases discovered for an entity $e \in \mathcal{E}_c$. For each phrase $h \in \mathcal{H}_e$ for an entity $e \in \mathcal{E}_c$, we compose a column vector $\mathbf{c}_h \in \mathbb{R}^n$ to represent the phrase $h$ as follows. The length of $\mathbf{c}_h$, i.e., $n$, is equal to the number of unique words and phrases available for the entities in $\mathcal{E}_c$. The locations of the nonzero elements of $\mathbf{c}_h$ are determined in the following manner. If $h$ is a single word, i.e., a degenerate phrase of length-1, then $\mathbf{c}_h$ has only one nonzero entry located at the corresponding location of the word $h$. If $h$ is a phrase with $q$ unique words where $q > 1$, then there are $(q + 1)$ nonzero entries in $\mathbf{c}_h$, where $q$ of the nonzero entries correspond to the $q$ unique words, respectively, and the one extra nonzero entry corresponds to the phrase as a whole. The values on these locations are the tf-idf weights derived from the bag of words and phrases model. All other locations have zero elements. In this formulation, the phrase both in its entirety as well as its individual components find respective weights in $\mathbf{c}_h$. Therefore, this formulation can confirm the occurrence of the phrase, and meanwhile has the ability to handle partial phrase match.

Next, we form a matrix $C \in \mathbb{R}^{n \times m}$ with each column $\mathbf{c}_h$ encoding a phrase $h$ for an entity in $\mathcal{E}_c$, in total $m$ unique phrases. To keep track of the mapping between the entity $e$ and the columns of $C$ that correspond to $e$'s phrases, we define $\mathcal{I}_e$ as the set of column indices corresponding to the phrases of entity $e$. Using the same vocabulary, we build a vector $\mathbf{d}$ to parameterize $D$. Note that a lookup table can be used to determine the lexical features appeared in $D$.

To see how the formulation can incorporate with the sparsity nature of SER discussed in Section 2.3, let us assume that $D$ describes an entity $e \in \mathcal{E}_c$. Ideally, a subset of the columns indexed by $\mathcal{I}_e$ in $C$, corresponding to $e$'s phrases which appear in $D$, should be highly correlated with $\mathbf{d}$. The other columns of $C$, which correspond to other entities or unmentioned aspects of $e$, may be merely correlated or completely uncorrelated with $\mathbf{d}$. Hence, we can model the correlation between the columns of $C$ and the web document $\mathbf{d}$ using the following linear model

$$\mathbf{d} = C\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^m$, and $\mathbf{n} \in \mathbb{R}^n$. Note that if $x_i \neq 0$, then $\mathbf{d}$ is correlated with $\mathbf{c}_i$, which means that the web document $D$ contains the phrase which is encoded in the column $i$ and belongs to some entity $e$, where $i \in \mathcal{I}_e$. If $x_i = 0$, then $\mathbf{d}$ is unrelated to column $\mathbf{c}_i$, which indicates phrase encoded in $\mathbf{c}_i$ does not appear or mostly irrelevant in the web document. Based on the sparsity nature discussed in Section 2.3, we assume that $\mathbf{x}$ is a *sparse* vector, which means most of the entries are zeros indicating no correlation between the document and the corresponding phrases. The sparse vector $\mathbf{x}$, especially its nonzero entries, contains critical information about the most discriminative lexical features in the web document which can help determine the entity. The vector $\mathbf{n}$ is used to model the noise, which absorbs the difference between the assumed model and the actual observed web document. To find $\mathbf{x}$, we leverage the idea behind the Lasso algorithm [31] and solve

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^m_+} \|\mathbf{d} - C\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{2}$$

where $\lambda > 0$ is the regularization parameter, and $\mathbb{R}^m_+$ denote the $m$-dimensional real space with nonnegative coefficients. We only focus on the $\mathbf{x}$ with nonnegative elements, since a negative coefficient does not permit an intuitive interpretation on the usage of the corresponding phrase. Note that the first term in (2) measures the quality of approximating the web document $\mathbf{d}$ using the $\ell_2$-norm distance, and the second term encourages a sparse $\hat{\mathbf{x}}$ to be learned.

The remaining task is to determine which entity is mentioned in $D$, or there is no such entity in $\mathcal{E}_c$. To this end, an additional step is required to process the resultant $\hat{\mathbf{x}}$. We adopt the following criterion. First, we find the entity $e^*$ as the maximizer to

$$\max_{e \in \mathcal{E}_c} \frac{\mathbf{d}^\intercal C \hat{\mathbf{x}}_{\mathcal{I}_e}}{\|\mathbf{d}\|_2 \cdot \|C\hat{\mathbf{x}}_{\mathcal{I}_e}\|_2} \equiv r^*$$

where $\mathbf{x}_{\mathcal{I}}$ denotes the vector by setting all entries of $\mathbf{x}$ to zero except those indexed by the elements in $\mathcal{I}$. As a result, the phrases associated with the entity $e^*$ can best approximate the web document measured by the cosine similarity defined in the Euclidean space. Next, we claim $e^*$ is the entity mentioned in $D$ if $r^* \geq \gamma$, where $\gamma \geq 0$ is a pre-defined parameter. Claim that there is no entity in $\mathcal{E}_c$ mentioned in $D$, i.e., an unknown entity, if $r^* < \gamma$.

Note that once we formulate a matrix $C$ as the parametrization of the knowledge base and transform the web document $D$ into a vector $\mathbf{d}$, many sparse signal recovery techniques apply immediately to finding $\hat{\mathbf{x}}$ according to the linear model (1), leading to a family of algorithms for SER using sparse signal recovery techniques.

### 2.5. Posterior Probability Pursuit

The techniques for sparse signal recovery commonly employ the $\ell_2$-norm distance as the metric of fitting quality [2, 31–33]. Note that the Bayesian formulations have proven effective for web information processing tasks [34]. In this section, we design a novel approach, namely Posterior Probability Pursuit (PPP), which equips with the negative log posterior probability as the distance metric, meanwhile taking advantage of the sparsity nature underlying this task by leveraging the ideas behind the sequential recovery algorithms [2, 4, 32].

We start with a preparatory setup. We formally define an *unknown* entity to represent any entity outside of the knowledge base, which is denoted by $e_u$. Let $\mathcal{F}_e$ denote the set of lexical features, including words and phrases, for entity $e$. For an entity $e \in \mathcal{E}$, we define $\mathsf{P}(f|e)$ as the probability of seeing the feature $f$ in the feature set $\mathcal{F}_e$. A typical approach for estimating $\mathsf{P}(f|e)$ is to apply the maximum likelihood estimate, which translates into the frequency of the occurrence of $f$ in $\mathcal{F}_e$. A smoothing method can be applied to improve the estimation [35]. For $e_u$, we define $\mathsf{P}(f|e_u) \triangleq \frac{|\{e: f \in \mathcal{F}_e\}|}{|\mathcal{E}|}$ for $f \in \cup_{e \in \mathcal{E}} \mathcal{F}_e$, which means that the probability of encountering feature $f$ for an unknown entity is approximated as the probability of seeing $f$ in the feature set of a random entity from the knowledge base. Let $\mathcal{G}$ be the set of lexical features found from the document $D$ against the confined entity set $\mathcal{E}_c$.

With this setup, we present the PPP algorithm below.

---

**Parameters:** $t \in (0, 1)$; $K, M \in \mathbb{N}$.

**Initialization:** $\mathcal{F}_0 = \emptyset, k = 1$.

**Step 1.** Let

$$f_k = \arg\min_{g \in \mathcal{G} \backslash \mathcal{F}_{k-1}} \left[ \min_{e \in \mathcal{E}_c} -\log \mathsf{P}(e|\mathcal{F}_{k-1} \cup \{g\}) \right] \quad (\star)$$

**Step 2.** Set $\mathcal{F}_k = \mathcal{F}_{k-1} \cup \{f_k\}$.

**Step 3.** Check the following termination criteria:
(1) $\max_{e \in \mathcal{E}_c} \mathsf{P}(e|\mathcal{F}_k) > t$. (2) $\mathcal{G} \backslash \mathcal{F}_k = \emptyset$. (3) $k = K$.
If any termination criterion is met, go to Step 4.
Otherwise, set $k \leftarrow k + 1$, and go to Step 1.

**Step 4.** Output: $\hat{e} = \arg\min_{e \in \mathcal{E}_c \cup \{e_u\}} -\log \mathsf{P}(e|\mathcal{F}_k)$ as the entity mentioned in $D$.

---

Note that to compute ($\star$), we use the following equations according to the Bayes rule and the conditional independence assumption

$$\mathsf{P}(e|\mathcal{F}) = \frac{\mathsf{P}(\mathcal{F}|e)\mathsf{P}(e)}{\sum_{e' \in \mathcal{E}_c \cup \{e_u\}} \mathsf{P}(\mathcal{F}|e')p(e')}$$

where

$$\mathsf{P}(e) = \frac{1}{M + |\mathcal{E}_c|}, \ \ e \in \mathcal{E}_c, \ \ \mathsf{P}(e_u) = \frac{M}{M + |\mathcal{E}_c|}$$
$$\mathsf{P}(\mathcal{F}|e) = \prod_{f \in \mathcal{F}} p(f|e), \ \ e \in \mathcal{E}_c \cup \{e_u\}.$$

The essence of the PPP algorithm is to find the best fit of an entity's semantics using a sparse set of lexical features from the web document where the fitting quality is measured by the posterior probability. To identify the sparse salient lexical features, PPP adopts the sequential selection principle, which is widely employed in matching pursuit algorithms [2, 32], to augment the set of salient features over a number of iterations. Particularly, in Step 1 of each iteration, the most discriminative and yet unselected lexical feature is found to maximally decrease the negative log posterior probability, which measures the agreement between an entity and the features selected from the web document thus far. By enforcing an early termination,

PPP encourages only a small amount of salient features to be sequentially selected for resolving the semantics, which echoes the sparsity nature underlying SER.

## 3. EXPERIMENT

We evaluate the proposed approaches for semantic entity resolution via the application of people entity resolution, where the goal is to determine whether some or none of the people entities in the knowledge base is mentioned in a given web document. The experimental setups are introduced as follows.

The knowledge base is built upon the information we crawled from a popular online social network. It contains the profiles of about 100 million unique people entities. Each profile employs the attribute-value pair representation to store various aspects of information about a person.

We randomly select 50 names from the people entity knowledge base. Using each selected name as a query, we obtain the top 20 results in the search engine result page via a commercial search engine. Then, the following types of web pages are manually removed: (i) Directory page. A web document of this kind usually mentions many different people entities with the same or similar names. (Common domains include http://pipl.com and http://www.spokeo.com.) (ii) Password-protected page. (iii) Web pages directly from the social network which are used to build our entity knowledge base. In total, we obtain 555 web documents for the 50 names. Note for each web document $D$, the people entities with the name which is used as the query to retrieve $D$ comprise the confined entity set $\mathcal{E}_c$. We manually determine for each web document the matching entity in $\mathcal{E}_c$, or claim that an unknown entity is mentioned.

We employ two baseline approaches for comparison, both based on the lexical features as discovered in Section 2.2. (i) *Cosine similarity with tf-idf weights* (tf-idf). The vector space representation of the web document and the people entity profiles are formed using the tf-idf weights. The cosine similarity is calculated to determine the best matching people entity. A threshold on the similarity is preset to determine the unknown people entity. This baseline method represents the lexical-features-based algorithmic component applied across a series of entity disambiguation techniques [21, 22, 29]. (ii) *Naive Bayes* (NB). A Naive Bayes classifier with the additional unknown entity is constructed. The people entity with the maximum posterior probability is claimed as mentioned in the web document. A pre-defined parameter is the total number of unknown population $M$, which equivalently specifies the prior probabilities of the entities. This baseline method essentially reproduces the state-of-the-art technique developed in [28] without the name variation model, which is learned with information beyond lexical features.

Define the following auxiliary quantities:

$n_1$ : the number of documents which the algorithm correctly determines the matching people entity in the knowledge base.

$n_2$ : the number of documents which the algorithm determines as mentioning some people entity in the knowledge base.

$n_3$ : the number of documents which has a matching people entity in the knowledge base, where $n_3 = 185$ in this experiment.

Then, we define the performance metrics as

$$\text{precision} = n_1/n_2, \quad \text{recall} = n_1/n_3.$$

For the Lasso based approach, there are extensive discussion on the parameter $\lambda$ [36]. An empirically choice is of the form $\lambda = \alpha\|C^{\mathsf{T}}\mathbf{d}\|_\infty$, where $\alpha = 0.01$ is a popular choice [37–39]. For PPP and NB, the parameter $M$, interpreted as the total amount of out-of-knowledge-base entities, determines the prior probabilities of all entities, which is difficult to estimate [40,41] and is beyond the scope of this paper. In order to fully understand the precision-recall tradeoff, we adopt a proper parameter grid (either one- or two-dimensional) and run the algorithm on all operating points. The performance tradeoffs are given in Fig.1. Note that for algorithm with more than one parameter, we only plot the best performance tradeoffs.

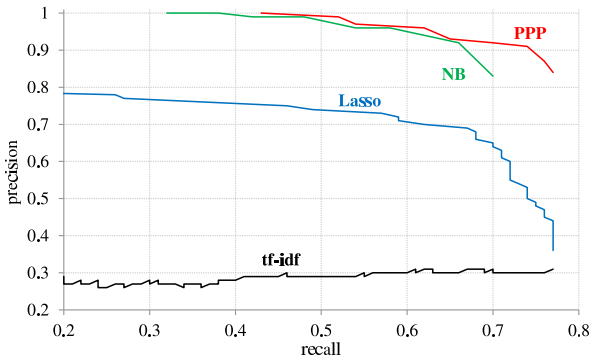


**Fig. 1**. Performance tradeoffs of algorithms. The parameter grids are as follows (step-sizes are omitted). tf-idf: threshold from 0 to 1. Lasso: $\alpha$ from $10^{-3}$ to 0.2, $\gamma$ from 0 to 1. NB: $M$ from $10^3$ to $10^{15}$. PPP: $t$ from 0.85 to $1 - 10^{-12}$, $M$ from $10^3$ to $10^{15}$, $K = 20$.

First, we compare the performance between tf-idf and the Lasso based approaches, since they both use $\ell_2$-norm based metrics. The difference is that while the tf-idf approach uses all the lexical features, the Lasso based approach only uses a sparse subset of lexical features. Clearly, the Lasso based approach delivers substantially improved precision over the tf-idf baseline at any given recall.

Next, we compare NB and PPP, since they both utilize the posterior probability as the metric. To enable thorough inspection, we detail their performances in the Table 1.

| | | PPP | | | | | | NB |
|---|---|---|---|---|---|---|---|---|
| $M$ | $t_0$ | 1 | 3 | 5 | 7 | 9 | 11 | |
| $10^8$ | P | 0.62 | 0.76 | 0.82 | **0.87** | **0.87** | **0.87** | 0.83 |
| | R | 0.79 | 0.78 | 0.77 | **0.76** | **0.73** | **0.73** | 0.70 |
| $10^{10}$ | P | 0.77 | 0.86 | 0.93 | 0.94 | 0.94 | **0.96** | 0.96 |
| | R | 0.66 | 0.65 | 0.65 | 0.63 | 0.62 | **0.62** | 0.58 |
| $10^{12}$ | P | 0.87 | 0.95 | 0.97 | 0.97 | **0.99** | **0.99** | 0.99 |
| | R | 0.57 | 0.56 | 0.54 | 0.52 | **0.52** | **0.52** | 0.48 |
| $10^{14}$ | P | 0.96 | 0.98 | 0.98 | **1** | **1** | **1** | 1 |
| | R | 0.47 | 0.44 | 0.44 | **0.43** | **0.43** | **0.42** | 0.38 |

**Table 1**. Comparison between PPP and NB. For PPP, $t = 1 - 10^{-t_0}$.

We can see from Table 1 that varying the parameters $M$ and $t$ in PPP leads to a tradeoff between precision (P) and recall (R). Note that for a fixed $M$, NB can be viewed as a special case of PPP by using all the lexical features, which can be algorithmically achieved by setting $t = 1$ (or, $t_0 = \infty$) and $K = \infty$ in PPP. From the P-R scores in boldface, PPP can substantially improve the recall by $7\% \sim 13\%$, at precisions no lower than its NB counterpart when $t$ is close to, but surely less than, one. Therefore, properly exploiting the sparsity nature enables the potential for performance improvement.

Overall, the PPP algorithm achieves the best precision-recall tradeoff among the techniques. Further, the techniques using posterior probability to measure the fitting quality substantially outperforms the techniques using $\ell_2$-norm based metrics, which indicates that the posterior probability is a more suitable metric for the entity-centric web information processing task.

## 4. REFERENCES


[1] I.F. Gorodnitsky, B. D. Rao, and J. George, "Source localization in magnetoencephalography using an iterative weighted minimum norm algorithm," *IEEE Asilomar Conf.*, 1992.

[2] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[3] S. Chen and D. Donoho, "Basis pursuit," Tech. Rep., Stanford Univ., 1995.

[4] J. A. Tropp, "Greedy is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[5] E. J. Candes, "Compressive sampling," *Proc. Int. Congr. Mathematicians*, 2006.

[6] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2009.

[7] D. Wipf and S. Nagarajan, "A unified bayesian framework for MEG/EEG source imaging," *NeuroImage*, pp. 947–966, 2008.

[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. PAMI*, vol. 31, no. 2, pp. 1–18, 2009.

[9] Y. Jin and B. D. Rao, "Algorithms for robust linear regression by exploiting the connection to sparse signal recovery," *Proc. ICASSP*, 2010.

[10] W. C. Chu, *Speech coding algorithms*, Wiley-InterSci., 2003.

[11] P.K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *Intl. Workshop Wearable and Implantable Body Sensor Networks*, 2009.

[12] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 8, pp. 508–518, 2000.

[13] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Communications*, vol. 50, pp. 374–377, 2002.

[14] D. Guo, "Neighbor discovery in ad hoc networks as a compressed sensing problem," *Proc. Inf. Th. Appl. Workshop*, 2009.

[15] C.D.Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, Cambrige Univ. Press, 2008.

[16] S. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani, "Emerging topic detection using dictionary learning," in *Proceedings of CIKM*, 2011.

[17] K. Min, Z. Zhang, J. Wright, and Y. Ma, "Decomposing background topics from keywords by principal component pursuit," in *Proceedings of CIKM*, 2010.

[18] Deepak Agarwal and Maxim Gurevich, "Fast top-k retrieval for model based recommendation," in *Proceedings of WSDM*, 2012.

[19] B. Hutchinson, M. Ostendorf, and M. Fazel, "A sparse plus low rank maximum entropy language model," in *Proceeding of InterSpeech*, 2012.

[20] Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, and A.Y. Ng, "Building high-level features using large scale unsupervised learning," in *Proc. of ICML*, 2012.

[21] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *Proc. of ACL*, 1998.

[22] R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proceedings of EACL*, 2006.

[23] X. Li, P. Morie, and D. Roth, "Identification and tracing of ambiguous names: discriminative and generative approaches," in *Proceedings of National Conf. AI*, 2004.

[24] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney, "Resolving surface forms to wikipedia topics," in *Proceedings of COLING*, 2010.

[25] D. Milne and I.H. Witten, "Learning to link with wikipedia," in *Proceedings of CIKM*, 2008.

[26] R. Mihalcea and A. Csomal, "Wikify! linking documents to encyclopedic knowledge," in *Proceedings of CIKM*, 2007.

[27] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *Proc. of EMNLP*, 2007.

[28] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proceedings of ACL-HLT*, June 2011, pp. 945–954.

[29] J. Hoffart, M. Amir Yosef, I. Bordino, H. Furstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *Proceedings of EMNLP*, 2011, pp. 782–792.

[30] K. Wang, N. Gloy, and X. Li, "Inferring search behaviors using partially observable markov model," in *Proc. of WSDM*, 2010.

[31] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[32] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

[33] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIREV*, vol. 43, no. 1, pp. 129–159, 2001.

[34] J. Lafferty and C. Zhai, "Probabilistic relevance models based on document and query generation," in *Language Modeling and Information Retrieval*. 2002, pp. 1–10, Kluwer Academic Publishers.

[35] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Harvard Univ., TR-10-98*, 1998.

[36] M. R. Osborne, B. Presnell, and B. A. Turlach, "On the lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 319–337, 1999.

[37] M. Figueiredo, Robert D. Nowak, and Stephen J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE JSTSP*, vol. 1, pp. 586–597, 2007.

[38] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE JSTSP*, vol. 1, pp. 606–617, 2007.

[39] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing," *Rice Univ., TR07-07*, 2007.

[40] I.J. Good, "The population frequency of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3, pp. 237–264, 1953.

[41] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acous. Sp. Sig. Proc.*, vol. 35, no. 3, pp. 400–401, 1987.