# Macau: Large-scale skill sense disambiguation in the online recruitment domain

**Conference Paper** · October 2015

DOI: 10.1109/BigData.2015.7363890

Faizan Javed
Home Depot

Ferosh Jacob
The Home Depot

# Macau: Large-Scale Skill Sense Disambiguation in the Online Recruitment Domain

**Qinlong Luo, Meng Zhao, Faizan Javed, Ferosh Jacob**

Data Science R & D, CareerBuilder.com

5550-A Peachtree Parkway, Norcross, GA 30092, USA

{Qinlong.Luo, Meng.Zhao, Faizan.Javed, Ferosh.Jacob }@CareerBuilder.com

*Abstract*—**Named entity sense disambiguation is a problem with important natural language processing applications. In the online recruitment industry, normalization and recognition of occupational skills play a key role in linking the right candidate with the right job. The disambiguation of multi-sense skills will help improve this normalization and recognition process. In this paper we discuss an automatic large-scale system to identify and disambiguate multi-sense skills, including: (1) Feature Selection: employing word embedding to quantify the skills and their contexts into vectors; (2) Clustering: applying Markov Chain Monte Carlo (MCMC) methods to aggregate vectors into clusters that represent respective senses; (3) Large-scale: implementing parallelization to process text blobs on a large-scale; (4) Pruning: cluster cleaning by analyzing intra-cluster cosine similarities. Based on experiments on sample datasets, the MCMC-based clustering algorithm outperforms other clustering algorithms for the disambiguation problem. Also based on data-driven in-house evaluations, our disambiguation system achieves 84% precision.**

***Keywords-named entity sense disambiguation; Markov Chain Monte Carlo; clustering; word embedding***

## I. INTRODUCTION

Ambiguity of word meanings is natural in most natural languages. As a classic example, the word *Apple* could refer to either *the firm* or *the fruit*. In addition, the level of ambiguity increases exponentially when short acronyms are referenced. For instance, *NLP* has at least two other common definitions: *Neuro-Linguistic Programming* and *NLP 5-step Sales Process*, beyond *Natural Language Processing*.

The R&D team at CareerBuilder (CB) has been working on developing solutions for recognizing and normalizing human capital entities (e.g., School, Company, Skill, Job title) [1, 2, 3]. Given proprietary taxonomies which we use in-house as knowledge bases, named entity sense disambiguation is critical to all the recognition and normalization tasks. We have received various customer feedbacks both internally and externally regarding instances of false positive tagging due to the ambiguity of skill entity senses. For instance, *SEARCH* has a minority sense in security/law enforcement in a text blob {*search, arrest, seizure*}, alongside a more common sense in the technology field, as shown in a text blob {*search, lucene, java*} [4]. Also *SERVER*, a common term in recruitment related documents, has both *food serving* and *information technology* senses. For accurate and relevant candidate and job matching, understanding the skill senses is a critical process in our recruitment ecosystem.

Disambiguation can be tackled by automatically assigning the most appropriate meaning to a multi-sense word with respect to a given context [5, 6, 7, 8, 9]. We encode the *context* of a *skill entity* by neural network-based language models [10, 11]. A *skill entity* is defined and identified by our taxonomy and tagging system [1] and its *context* refers to text blobs containing at least one mention of that *skill entity*. *Contexts* are extracted from either skill section(s) of *resumes* or requirement section(s) of *job postings*. We used a popular tool *word2vec* [12] with parameter adjustments tailored for our data to semantically quantify *contexts*. For each and every *skill entity* from our taxonomy, we use the average of *word2vec* vectors of its *context* to represent the corresponding sense, if any. See [1] for an overview of how *word2vec* is customized and utilized in-house. Once we obtain a vector representation of the *context* of a *skill entity*, we treat such vector as an *equivalent* representation of a potential skill sense. Those vectors are then used as input for clustering algorithms such that in each cluster, *contexts* (represented by vectors) aggregated could be used to denote a *skill sense*.

Successful disambiguation systems either make use of public knowledge bases [13, 14, 15], or are data-driven [16, 17, 18], to the best of our knowledge. For disambiguation of occupational skills, we have observed that results from common knowledge bases like Wikipedia are not domain specific enough, as such possess often inherits a certain degree of ambiguity that requires further investigation. On the other hand, efficient data-driven systems are mostly developed using supervised learning methods that induce daunting annotation costs.

Inspired by two pioneering publications [19, 20], we adopt the Monte Carlo Markov Chain (MCMC) concept as an unsupervised clustering method. MCMC is applicable for our disambiguation task, arguably, because the *word2vec* quantification of *contexts* of a given *skill entity* is readily available as a unique feature in our system. Such quantification through word embedding enables us to encode semantic relationships that represent senses into mathematical formulations.

In this paper we propose Macau, a large scale skill entity sense disambiguation framework based on MCMC-based clustering algorithm. The system processes hundreds of millions of documents to identify various skill senses, without the requirement of pre-selecting total number of clusters as opposed to k-means [21] or Latent Dirichlet Allocation (LDA) [22]. To counter the computational cost of MCMC, we use

CARBi [23], our in-house Big Data framework built over Twitter's Scalding[1].

The remainder of the paper is structured as follows. Macau, our MCMC-based disambiguation system is discussed in Section II. In Section III, we compare our system with other unsupervised learning approaches commonly employed in the field. Section IV provides a thorough review of related research in the topic of both word and named entity sense disambiguation. We conclude our work with discussion and future work in Section V.

## II. Methodology

### A. *Mathematical Model*

The skills disambiguation system is designed to disambiguate multi-sense skills based on corresponding contexts. For a specific ambiguous skill we need to:

1) Determine the number of senses of the ambiguous skill.

2) Retrieve related information (such as related skill terms) to represent the different senses.

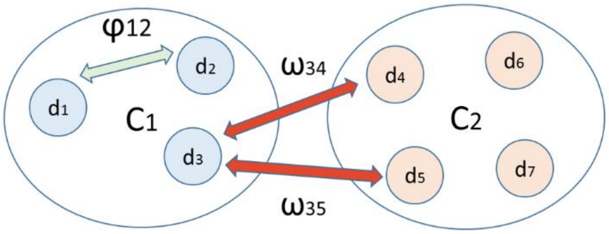

Figure 1. Illustration of correlations between documents, where the small orange circles represent documents and large blue circles represent clusters. Green and red arrows label affinity and repulstion factors, respectively.

Figure 1 demonstrates the initialization of the disambiguation system. Each document $d_i$ contains a set of skill surface forms extracted from a recruitment document (e.g., skills sections of resumes or job requirements sections of job ads). Each cluster $C_i$ is a collection of documents and represents one unique sense of the ambiguous skill. The key process to solve this problem is to group the documents into different clusters, such that each cluster contains "similar" documents. The skill is ambiguous if two or more clusters are generated by the clustering process. After clustering, we can extract the related information from the clusters to represent the senses. One simple strategy is to count the frequency of the skill surface forms in a cluster and the high frequent ones can be used to represent the cluster (sense).

To solve this clustering problem, an objective function is introduced:

$$f(\overleftrightarrow{C}) = \sum_{i=0}^{k} \left( \sum_{m,n\in C_i,\, m\neq n} \varphi_{mn} + \sum_{m\in C_i,\, n\in \cup_{j\neq i} C_j} \omega_{mn} \right) \quad (1)$$

where $k$ is the number of clusters and $\overleftrightarrow{C}$ is the clustering states, which represents the document-cluster relationship.

In Equation (1), the affinity factors and repulsion factors are defined as:

$$\varphi_{mn} = \cos(v_m, v_n) - b \quad (2)$$

$$\omega_{mn} = -[\cos(v_m, v_n) - b] \quad (3)$$

respectively, where the cosine similarity function is given by:

$$\cos(v_i, v_j) = v_i \times v_j / \|v_i\| \cdot \|v_j\| \quad (4)$$

and the parameter $b$ is the bias term.

In Figure 1, green arrows represent affinity factors, which measure how "similar" the two documents in the same cluster are. Red arrows represent repulsion factors, which measure how "different" two documents are in separated clusters. Equation (1) defines a function mapping the correlations between all the documents to a single value which measures how good the clustering result is. Theoretically, this objective function should be at the maximum value, if the clustering is perfect. As in most optimization based learning algorithms, the optimal clustering of documents, denoted by $\overleftrightarrow{C}_{optimal}$, is the one that maximizes the cost function $f(\overleftrightarrow{C})$, such that

$$\overleftrightarrow{C}_{optimal} = arg\max_{\overleftrightarrow{C}} f(\overleftrightarrow{C}) \quad (5)$$

Equation (5) serves as the mathematical foundation to transform clustering task into searching for the global maximum of the cost function.

### B. *Word Embedding*

As discussed in [1], we utilized a unique feature of our recruitment data for taxonomy building and skills tagging: people tend to list related skills together in the skills sections of resumes or requirements sections of job postings. We consequently consider it a reasonable assumption that senses of a given skill are embedded in the context where the skill is listed. For example, if a candidate lists *SQL*, *SSIS*, and *SSAS* together with *server*, we can assume the person meant *server* in the *IT* sense. Likewise, if *bartending*, *food*, *restaurant* are documented together with *server*, then we are safe to understand the sense as *food serving*.

Given a skill document, we encode the context using the n-gram model. We then use neural networks to reflect possible semantic relevance with contexts of the given document and other documents in the corpus. To train *word2vec* vectors for the entire corpus (resumes/job postings), we chose the *skip-gram* model with *hierarchical softmax* [10], allowing random down-sampling of high frequency terms [12]. The model setting is selected for best training results empirically [1]. For computational efficiency, we constructed n-grams *a priori* utilizing the unique structure of our human capital data and trained a unigram model on the

[1] Scalding, https://twitter.com/scalding

corpus, by replacing spaces ('\s+') by underscores ('_'). To compute a vector representation of the document (aka, skills/job requirement sections) of a given skill mention, we use the average of *word2vec* vectors of all the terms present in the document. Those vectors are thereafter employed for clustering based on MCMC techniques.

### C. Markov Chain Monte Carlo

TABLE I. MCMC-BASED CLUSTERING ALGORITHM

```
1. Randomly choose the initial distribution $\overleftrightarrow{C}_0$
2. For i = 1 to N
   2.1 Propose a change $\overleftrightarrow{C}$ randomly, from $\overleftrightarrow{C}_{i-1}$ to $\overleftrightarrow{C}_i^*$
   2.2 Calculate the change of the cost function:
       $\Delta f(\overleftrightarrow{C}) = f(\overleftrightarrow{C}_i^*) - f(\overleftrightarrow{C}_{i-1})$
   2.3 Calculate transition probability with exponential transformation:
       $p = \exp(\frac{-\Delta f(\overleftrightarrow{C})}{T})$
   2.4 Generate random number $r \sim U[0,1)$
   2.5 Compare $p$ and $r$:
       if $p > r$, accept the proposed change, $\overleftrightarrow{C}_i = \overleftrightarrow{C}_i^*$;
       else, reject the proposed change, $\overleftrightarrow{C}_i = \overleftrightarrow{C}_{i-1}$
3. For j = 1 to M
   3.1 repeat the same steps as in 2.1 ~ 2.5
   3.2 save the state $\overleftrightarrow{C}_j$ after each iteration
4. The final solution $\overleftrightarrow{C}_{optimal}$ is the average value of $\overleftrightarrow{C}_j$
       $\overleftrightarrow{C}_{optimal} = \frac{1}{M}\sum_{j=1}^{M}\overleftrightarrow{C}_j$
```

MCMC techniques are widely used for optimization in Physics and Social Science. In this work, MCMC, more specifically, the Metropolis-Hastings (MH) algorithm and Simulated Annealing (SA) [24, 25] method, are employed as an optimization algorithm to maximizing $f(\overleftrightarrow{C})$. One of the primary advantages of MCMC-based algorithms is that, given sufficient iteration steps, they can provide a good approximation to the global optimum of a complex function, compared to other iterative optimization approaches [24, 25, 26]. Details of our MCMC-based optimization process are shown in Table I.

### D. Distributed System for Big Data

By the nature of most MCMC-based algorithms, their implementation is normally computationally intensive and expensive due to massive calculations of $\Delta f(\overleftrightarrow{C})$, the change of the objective function. Therefore, this sequential MCMC process is not applicable for a large number of documents.

We use a distributed system to scale Macau by processing input documents in parallel. A dataset containing $N$ documents, is first equally divided into $k$ partitions. These partitions will be distributed to different nodes and MCMC process will be applied to each of these partitions simultaneously. As a result, each MCMC process will produce $C_i$ clusters. These clusters will be collected together and each cluster is mathematically represented by the centroid of the documents in that cluster. As a final step, MCMC clustering will be applied again to all the transformed documents (clusters) producing the final clusters.

In terms of speed, the distributed system will accelerate the clustering process dramatically due to the fact that k partitions are processed in parallel. Besides the fast speed, this distributed system can be easily extended to three or more cascading steps, from current two cascading steps. This distributed system is implemented using CARBi, our in-house solution for Big Data applications. The application was executed in CB Hadoop cluster (50 nodes, 210 cores). All the experimental results are recorded after executing in the CB cluster with 25% occupancy.

### E. Pruning

In this skills sense disambiguation problem, we use resumes from CB's internal database which contain various noises. For instance, some job seekers may put some irrelevant skills on their resumes, in order to gain more attentions. From our observations, we found that most of the clusters contain some noisy skill terms. In order to make the final clusters as clean as possible, a pruning method is applied to clean the resulting clusters from MCMC clustering step.

Given a cluster, the proposed method is comprised of three steps:

1) Calculate cosine similarities (see Equation (4)) between all documents within the cluster;

2) For each document, calculate the average cosine similarity between this document and other documents in the same cluster;

3) Remove the document if the average similarity falls below a pre-defined threshold.

The goal is to minimize intra-cluster distance to ensure enough saturation of the clustering method. Note that the embedding method employed in our system will ensure the cosine distance metric is sufficient to measure semantic closeness.

## III. EVALUATION AND DISCUSSION

Macau, our MCMC-based clustering algorithm is evaluated in several different ways:

1) Macau is compared with other popular clustering algorithms such as k-means and LDA in Section A.

2) The distributed system is compared to the sequential system in terms of accuracy and speed in Section B.

3) The proposed pruning algorithm shows the improvements of the results in Section C.

4) Skills tagging with disambiguation is evaluated by a data-drive approach in Section D.

### A. Macau vs. k-means vs. LDA

Skill sense disambiguation is essentially a clustering problem and thus can be solved by other means of clustering algorithms. We compare Macau with both classic and modern alternatives, so we choose k-means and LDA as baselines.

A methodological comparison of the clustering algorithms is performed using three datasets:

1) “SEARCH” dataset that has 500 documents that contain the ambiguous skill “search”;

2) “SERVER” dataset that has 1000 documents that contain the ambiguous skill “server”;

3) “NLP” dataset that has 259 documents that contain the ambiguous skill “NLP”.

These datasets are manually labeled beforehand. The “SEARCH” and “SERVER” datasets both contain 2 different senses while the “NLP” dataset has 3 different senses.

In this comparison, vectors from *word2vec* [12] (same as used in Macau) are used for k-means clustering and the pruning method is not applied to Macau. We applied on-line LDA [27] through *Vowpal Wabbit* [28] for LDA.

Figure 2 shows the accuracy of the clustering results from the three methods mentioned above. Here accuracy is defined as the percentage of the documents that are assigned to the right clusters over all the documents. From the experimental results of the three datasets, Macau outperforms k-means and LDA in general.

In Macau, the number of clusters is not required to be pre-specified, but instead, will be learned automatically. This number can be as small as two, or it can be as large as the number of documents, which means each document represent one unique sense at the beginning. Generally speaking, it is only required to specify the maximum number of clusters at the beginning, which is the number of documents. In order to compare with k-means and LDA where the number of clusters is required, the maximum number of clusters is set to be $2, 3, 5, 10$ in Macau.

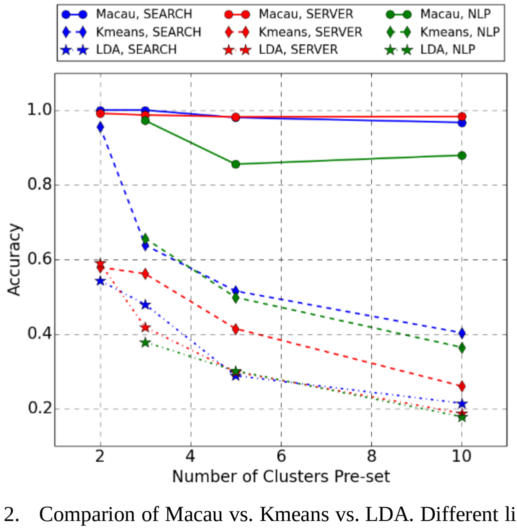

Figure 2. Comparion of Macau vs. Kmeans vs. LDA. Different line styles represent different algorithms, and different color (blue, red, green) represents the results for disambiguating “SEARCH”, “SERVER” and “NLP” respectively. The number of clusters for “NLP” are not pre-set to 2 because it has three different senses.

Across all three datasets in Figure 2, accuracies of k-means and LDA depreciate as cluster numbers pre-set increases, due to relatively small amount of actual senses represented in the testing sets. However the accuracy of Macau persists, almost independently of number of cluster pre-defined. Even if pre-specified amount clusters considerably outweighs the actual amount of skill senses. Macau will always converge to the desired cluster distributions, demonstrating great scale of availability and stability. Since both Macau and k-means take *word2vec* vectors as input, the difference lies in the algorithms themselves.

One other advantage of Macau is the tolerance of empty clusters, since the mathematical proposed change is in reality to move one document from one cluster to another cluster. Whereas for k-means, a null cluster is not algorithmically permitted at all. The fact that pre-determined value for the number of clusters is not required in Macau ensures reliable and consistent performance without human intervention. This is also one of the reasons for us to favor Macau over k-means or LDA for this disambiguation problem because the number of potential senses for a given skill entity is generally unknown *a priori*.

Figure 2 also shows the relatively low performance of LDA for disambiguation. We argue that LDA is not a reasonable choice for due to the primary differences between LDA and MCMC/k-means are:

1) Unlike MCMC and k-means that assign each document to a single cluster, LDA can assign different topics to a document with different probabilities. In other words, a document will be assigned to different topics concurrently. Therefore simply choosing the topic with highest probability may not necessarily represent the reality.

2) In our specific disambiguation challenge, “topic” is not necessarily equivalent to “word sense”, e.g. one sense may contain several different topics or one topic may cover several different senses. It is virtually not possible to figure out which topic belongs to which sense without human curation.

### B. *Distributed Macau System*

In Macau, most of the computation cost is incurred by the calculation of the change of the cost function $\Delta f(\overleftrightarrow{C})$. To be more specific, the execution time of calculating $f(\overleftrightarrow{C})$ is $O(n^2)$, according to the definition of the cost function in Equation (1). However, as shown in Figure 3 by the blue dot-curve, the CPU time of sequential Macau goes up linearly, namely in the order of $n$. The reason is that we calculate $\Delta f(\overleftrightarrow{C})$ directly, which costs $O(n)$ time. This is our challenge to scale up the sequential MCMC system.

Figure 3 shows the comparison of the distributed system and sequential system in terms of speed and accuracy. In Figure 3 (a), the blue curve shows that CPU time of sequential system goes up linearly as the number of documents increases. However, the red curve shows clearly that the CPU time of distributed system stays almost unchanged with the increase of the number of documents, which provides a clear evidence of the advantage of the distributed system. When the dataset size is 1000, the difference of CPU time is 13 min (distributed) vs. 57 min (sequential). For Big Data applications as in most of our use cases, we can always keep the running time constant, if enough nodes can be deployed on demand.

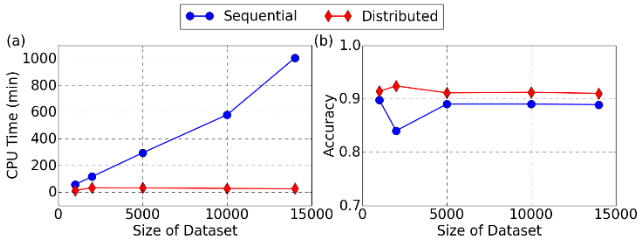

Figure 3. Comparion between distributed system and sequential system: (a) CPU time vs. dataset size; (b) Accuracy vs. dataset size. The skill term "SERVER" is used in this experiment. In the distributed MCMC system, each partition contains 100 documents.

Generally speaking, accuracy will be sacrificed to improve the speed. However, Figure 3 (b) provides the evidence that the distributed system has similar accuracy as the sequential system. The accuracy of the distributed system retains above 90% with the increase of the data size. The key point is that this MCMC clustering algorithm can be scaled up with very little compromise in accuracy which is another primary advantage of this distributed system.

### C. *Pruning*

In order to emphasize the difference before and after pruning, the same datasets ("SEARCH", "SERVER" and "NLP") are used for evaluation, as shown in Table II. It can be seen that the accuracy increases about 4% in average after applying pruning to Macau.

TABLE II. ACCURACY OF 3 DATASETS BEFORE AND AFTER PRUNING

| **Table Head** | **Dataset** | | |
|---|---|---|---|
| | ***SEARCH*** | ***SERVER*** | ***NLP*** |
| No pruning | 92.8% | 98.3% | 87.9% |
| Pruning | 96.7% | 90.0% | 56.1% |

To use this pruning method, we need to be careful of the fact that it may deplete the entire cluster if the pre-determined threshold is not met by any document. Therefore, if the dataset is too small or the data is too noisy, this pruning method is used with caution for the balance of availability and precision. Since it is a general-purpose technique, this pruning method can be also be applied to other vector-based clustering algorithms, such as k-means.

### D. *Data-Driven disambiguation Evaluation*

We evaluated Macau through a data-driven approach. The goal is to evaluate the accuracy of the sense disambiguation for skills tagging.

More than 29k randomly selected resumes were tagged, covering 90% of all ambiguous skills suggested by the system after distributed MCMC clustering. 1% of resumes of *each* ambiguous skill were selected as a starting point for evaluation [14] because no accuracy expectation was assumed, resulting in about 400 resumes. Samples were expanded incrementally with 1% of total population each time until sample size was saturated and a conclusion could be made with a desired confidence level. Three human evaluators were asked to evaluate the legitimacy and relevancy of tagging results, with respect to the input text. Evaluations are aggregated and consolidated afterward.

Regarding precision as percent of right senses tagged for the right contexts, we are able to conclude at 95% confidence level that the aggregated precision of our system tops 84%, based on the formulations below.

## IV. RELATED WORK

There is a long history of work in the disambiguation area as part of the Natural Language Processing (NLP) and Machine Translation (MT) communities [8]. The proliferation of Big Data in semantic and social web has resulted in an increased interest in better and more scalable approaches for disambiguation. One of the early systems for large-scale disambiguation was presented in [13]. This system leverages Wikipedia to perform entity identification and disambiguation. Contextual and category information such as surface forms, category tags and contexts are constructed using entity pages, disambiguation pages and redirect pages. This information is used to construct a vector space model that is used in a disambiguation model. A clustering-based approach for named-entity disambiguation for the Web at web-scale is presented in [6]. The approach assumes that mentions of an entity can be disambiguated using information about the entities with which it co-occurs within the same document. The well-known Lesk disambiguation algorithm [29] is an overlap-based algorithm that exploits the number of common words in two sense definitions (aka glosses) to select the proper meaning in a context. More recently the Lesk algorithm was extended with a distributional semantic model (DSM) that replaces the concept of overlap with that of similarity [5]. AGDISTIS [7] uses a disambiguation graph along with the HITS algorithm [30] for knowledge-base independent named-entity disambiguation that scales to large knowledge bases.

For unsupervised clustering problems, MCMC has been demonstrated to converge to global minima, especially in situations where the clusters have strong overlap and centroid oriented methods fail [26]. In [19] the related problem of cross-document co-reference resolution is tackled by representing the problem domain as an undirected graphical model and applying a distributed version of a MCMC-based inference technique to group mentions into distinct entities. MCMC has also been used as a clustering algorithm in a parallelized SQL implementation for clustering in large database management systems [31].

## V. CONCLUSION AND FUTURE WORKS

In this work, Macau, a large-scale MCMC-based clustering algorithm is introduced to solve the disambiguation problem for our normalized skill terms. It shows that Macau has very high accuracy and we do not need to pre-set the number of clusters explicitly. Therefore, this method is preferred over the other clustering algorithms such as k-means and LDA. For large-scale datasets, a general purpose distributed system based on MCMC clustering algorithm is also proposed to parallelize the clustering process. This highly extendable system can be applied to other clustering algorithms easily. Finally, a data-driven evaluation has been conducted to investigate the performance of Macau by using real-world data.

This work is based on the MCMC clustering algorithm without optimization. In the future, we are planning to optimize the MCMC clustering algorithm, e.g. automating the annealing steps or inserting more terms to the objective function to filter the noisy data out. Since one of the disadvantage of k-means is to choose the number of clustering beforehand, defining a cost function for k-means to auto assign the number of "k", is the other way we may go in the future.

## References

[1] M.Zhao, F.Javed, F.Jacob and M.McNair, "SKILL: A System for Skill Identification and Normalization," Proc. the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence (IAAI 2015), AAAI Press, Jan. 2015, in press.

[2] F. Javed, Q. Luo, M. McNair, F. Jacob, M. Zhao, and T. Kang, "Carotene: A Job Title Classification System for the Online Recruitment Domain" in Proceedings of the International Conference of Big Data Computing Service and Applications, San Francisco Bay, CA, April 2015.

[3] F. Jacob , F. Javed, M. Zhao, and M. Mcnair, "sCooL: A System for Academic Institution Name Normalization" in Proceedings of the Second International Symposium on Big Data and Data Analytics, Minneapolis, MN, May 2014.

[4] M.Bastian, M.Hayes, W.Vaughan, S.Shah, P.Skomoroch and H.Kim, "LinkedIn Skills: Large-Scale Topic Extraction and Inference," Proc. the 8th ACM Conference on Recommender Systems (RecSys 2014), ACM, Oct. 2014, pp. 1-8, doi: 10.1145/2645710.2645729.

[5] P.Basile, A.Caputo and G.Semeraro, "An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model," Proc. the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014), ACL Press, Aug. 2014, pp. 1591-1600.

[6] L. Sarmento, A. Kehlenbeck, E.Oliveira and L.Ungar, "An Approach to Web-scale Named-Entity Disambiguation," Lecture Notes in Computer Science, vol. 5632, Springer, 2009, pp. 689-703.

[7] R.Usbeck, A. Ngomo, M.Röder, D.Gerber, S.Coelho and S.Auer, "AGDISTIS - Graph-Based Disambiguation of Named Entities using Linked Data," The Semantic Web – ISWC 2014, vol. 8796, Lecture Notes in Computer Science, International Publishing: Springer, Oct. 2014, pp. 457-471.

[8] R. Navigli, "Word sense disambiguation: A Survey," ACM Computing Surveys, vol. 41, Feb 2009, pp. 1-69, doi: 10.1145/1459352.1459355.

[9] C. Manning and H. Schütze, Statistical Natural Language Processing, MIT Press, 1999, pp. 230-261.

[10] Y. Bengio, R.Ducharme, P.Vincent and C.Jauvin, "A Neural Probabilistic Language Model," Journal of Machine Learning Research, vol. 3, Feb. 2003, pp. 1137-1155.

[11] T. Milolov, A. Deoras, D. Povey, L. Burget and J. Cernocký, "Strategies for Training Large Scale Neural Network Language Models," Proc. IEEE Automatic Speech Recognition and Understanding Workshop, IEEE Press, Dec 2011, pp. 196-201, doi: 10.1109/ASRU.2011.6163930.

[12] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Proc. International Conference on Learning Representations, May 2013.

[13] S. Cucerzan, "Large-Scale Named Entity Disambiguation Based on Wikipedia Data," Proc. Joint Conference on Empirical Models in Natural Language Processing and Computational Natural Language Learning, ACL Press, June 2007, pp. 708-716.

[14] B.Dandala, R.Mihalcea and R.Bunescu, "Multilingual Word Sense Disambiguation Using Wikipedia," Proc. Sixth International Joint Conference on Natural Language Processing (IJCNLP 2013), Dec. 2013, pp. 498-506.

[15] M. Galley and K. McKeown, "Improving Word Sense Disambiguation in Lexical Chaining," Proc. the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), 2003.

[16] H.T. Ng and H.B. Lee, "Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Examplar-Based Approach," Proc. the 34th Annual Meeting of the Association for Computational Linguistics, ACL Press, 1996.

[17] T. Pedersen, "A Decision Tree of Bigrams is An Accurate Predictor of Word Sense," Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL 2001), June 2001, pp. 79-86.

[18] W. Gale, K. Church, and D. Yarowsky, "A method for disambiguating word senses in a large corpus," Computers and the Humanities, 1992, vol. 26, pp. 415-439.

[19] S. Singh, A. Subramanya, F. Pereira and A. McCallum, "Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models," Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (HLT 2011), ACL Press, 2011, pp. 793-803.

[20] S. Singh, M. Wick and A. McCallum, "Monte Carlo MCMC: efficient inference by approximate sampling," Proc. Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL 2012), ACL Press, July 2012, pp. 1104-1113.

[21] J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations," Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, pp. 281-297.

[22] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation" in Journal of Machine Learning Research, **3** (4-5), 2003, pp. 993-1022.

[23] F. Jacob, A. Johnson, F. Javed, M. Zhao and M. McNair, "WebScalding: A Framework for Big Data Web Services," Proc. International Conference of Big Data Computing Service and Applications, Apr. 2015, in press.

[24] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by Simulated Annealing", Science, 220 (4598), pp. 671-680, 1983.

[25] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", Journal of Optimization Theory and Applications, 45(1), pp. 41-51, 1985.

[26] B. D. Stošić, "Pairwise Clustering Using A Monte Carlo Markov Chain", Physica A: Statistical Mechanics and its Applications, 388(12), 2009, pp. 2373-2382.

[27] M. Hoffman, D. Blei, and F. Bach, "Online Learning for Latent Dirichlet Allocation" in Advances in Neural Information Processing Systems (NIPS), Vancouver, 2010.

[28] A.Agarwal, O. Chapelle, M. Dudik, and J. Langford, "A Reliable Effective Terascale Linear Learning System", Journal of Machine Learning Research, vol. 15, Mar. 2014, pp. 1111-1133.

[29] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell A Pine Cone From An Ice Cream Cone," In Proceedings of the 5th Annual International Conference on Systems Documentation, ACM SIGDOC, 1986, pp. 24-26, New York, NY, USA.

[30] J. M. Kleingberg, "Authoritative Sources in A Hyperlinked Environment", J. ACM, 46(5), pp. 604-632, 1999.

[31] D. S. Matusevich and C. Ordonez, "A Clustering Algorithm Merging MCMC and EM Methods Using SQL Queries", In Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, 2014, pp. 61-76.