# Learning the Latent Semantics of a Concept from its Definition

**Weiwei Guo**
Department of Computer Science,
Columbia University,
New York, NY, USA
weiwei@cs.columbia.edu

**Mona Diab**
Center for Computational Learning Systems,
Columbia University,
New York, NY, USA
mdiab@ccls.columbia.edu

## Abstract

In this paper we study unsupervised word sense disambiguation (WSD) based on sense definition. We learn low-dimensional latent semantic vectors of concept definitions to construct a more robust sense similarity measure *wmfvec*. Experiments on four all-words WSD data sets show significant improvement over the baseline WSD systems and LDA based similarity measures, achieving results comparable to state of the art WSD systems.

## 1 Introduction

To date, many unsupervised WSD systems rely on a sense similarity module that returns a similarity score given two senses. Many similarity measures use the taxonomy structure of WordNet [WN] (Fellbaum, 1998), which allows only noun-noun and verb-verb pair similarity computation since the other parts of speech (adjectives and adverbs) do not have a taxonomic representation structure. For example, the *jcn* similarity measure (Jiang and Conrath, 1997) computes the sense pair similarity score based on the information content of three senses: the two senses and their least common subsumer in the noun/verb hierarchy.

The most popular sense similarity measure is the Extended Lesk [*elesk*] measure (Banerjee and Pedersen, 2003). In *elesk*, the similarity score is computed based on the length of overlapping words/phrases between two extended dictionary definitions. The definitions are extended by definitions of neighbor senses to discover more overlapping words. However, exact word matching is lossy. Below are two definitions from WN:

**bank#n#1:** *a financial institution that accepts deposits and channels the money into lending activities*
**stock#n#1:** *the capital raised by a corporation through the issue of shares entitling holders to an ownership interest (equity)*

Despite the high semantic relatedness of the two senses, the overlapping words in the two definitions are only *a, the*, leading to a very low similarity score.

Accordingly we are interested in extracting latent semantics from sense definitions to improve *elesk*. However, the challenge lies in that sense definitions are typically too short/sparse for latent variable models to learn accurate semantics, since these models are designed for long documents. For example, topic models such as LDA (Blei et al., 2003), can only find the dominant topic based on the observed words in a definition ($financial$ topic in $bank\#n\#1$ and $stock\#n\#1$) without further discernibility. In this case, many senses will share the same latent semantics profile, as long as they are in the same topic/domain.

To solve the sparsity issue we use missing words as negative evidence of latent semantics, as in (Guo and Diab, 2012). We define missing words of a sense definition as the whole vocabulary in a corpus minus the observed words in the sense definition. Since observed words in definitions are too few to reveal the semantics of senses, missing words can be used to tell the model what the definition is **not** about. Therefore, we want to find a latent semantics profile that is related to observed words in a definition, but also **not** related to missing words, so that the induced latent semantics is unique for the sense.

Finally we also show how to use WN neighbor sense definitions to construct a nuanced sense similarity *wmfvec*, based on the inferred latent semantic vectors of senses. We show that *wmfvec* outperforms *elesk* and LDA based approaches in four All-words WSD data sets. To our best knowledge, *wmfvec* is the first sense similarity measure based on latent semantics of sense definitions.

*Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 140–144,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

| | financial | sport | institution | $R_o$ | $R_m$ |
|---|---|---|---|---|---|
| $v_1$ | 1 | 0 | 0 | 20 | 600 |
| $v_2$ | 0.6 | 0 | 0.1 | 18 | 300 |
| $v_3$ | 0.2 | 0.3 | 0.2 | 5 | 100 |

Table 1: Three possible hypotheses of latent vectors for the definition of $bank\#n\#1$

## 2 Learning Latent Semantics of Definitions

### 2.1 Intuition

Given only a few observed words in a definition, there are many hypotheses of latent vectors that are highly related to the observed words. Therefore, missing words can be used to prune the hypotheses that are also highly related to the missing words.

Consider the hypotheses of latent vectors in table 1 for $bank\#n\#1$. Assume there are 3 dimensions in our latent model: *financial, sport, institution*. We use $R_o^v$ to denote the sum of relatedness between latent vector $v$ and all observed words; similarly, $R_m^v$ is the sum of relatedness between the vector $v$ and all missing words. Hypothesis $v_1$ is given by topic models, where only the $financial$ dimension is found, and it has the maximum relatedness to observed words in $bank\#n\#1$ definition $R_o^{v_1} = 20$. $v_2$ is the ideal latent vector, since it also detects that $bank\#n\#1$ is related to $institution$. It has a slightly smaller $R_o^{v_2} = 18$, but more importantly, its relatedness to missing words, $R_m^{v_2} = 300$, is substantially smaller than $R_m^{v_1} = 600$.

However, we cannot simply choose a hypothesis with the maximum $R_o - R_m$ value, since $v_3$, which is clearly not related to $bank\#n\#1$ but with a minimum $R_m = 100$, will therefore be (erroneously) returned as the answer. The solution is straightforward: give a smaller weight to missing words, e.g., so that the algorithm tries to select a hypothesis with maximum value of $R_o - 0.01 \times R_m$. We choose weighted matrix factorization [WMF] (Srebro and Jaakkola, 2003) to implement this idea.

### 2.2 Modeling Missing Words by Weighted Matrix Factorization

We represent the corpus of WN definitions as an $M \times N$ matrix $X$, where row entries are $M$ unique words existing in WN definitions, and columns represent $N$ WN sense ids. The cell $X_{ij}$ records the TF-IDF value of word $w_i$ appearing in definition of sense $s_j$.

In WMF, the original matrix $X$ is factorized into two matrices such that $X \approx P^\top Q$, where $P$ is a $K \times M$ matrix, and $Q$ is a $K \times N$ matrix. In this scenario, the latent semantics of each word $w_i$ or sense $s_j$ is represented as a $K$-dimension vector $P_{\cdot,i}$ or $Q_{\cdot,j}$ respectively. Note that the inner product of $P_{\cdot,i}$ and $Q_{\cdot,j}$ is used to approximate the semantic relatedness of word $w_i$ and definition of sense $s_j$: $X_{ij} \approx P_{\cdot,i} \cdot Q_{\cdot,j}$.

In WMF each cell is associated with a weight, so missing words cells ($X_{ij}$=0) can have a much less contribution than observed words. Assume $w_m$ is the weight for missing words cells. The latent vectors of words $P$ and senses $Q$ are estimated by minimizing the objective function:[1]

$$\sum_i \sum_j W_{ij} \left(P_{\cdot,i} \cdot Q_{\cdot,j} - X_{ij}\right)^2 + \lambda||P||_2^2 + \lambda||Q||_2^2$$

$$\text{where } W_{i,j} = \begin{cases} 1, & \text{if } X_{ij} \neq 0 \\ w_m, & \text{if } X_{ij} = 0 \end{cases} \qquad (1)$$

Equation 1 explicitly requires the latent vector of sense $Q_{\cdot,j}$ to be not related to missing words ($P_{\cdot,i} \cdot Q_{\cdot,j}$ should be close to 0 for missing words $X_{ij} = 0$). Also weight $w_m$ for missing words is very small to make sure latent vectors such as $v_3$ in table 1 will not be chosen. In experiments we set $w_m = 0.01$.

After we run WMF on the definitions corpus, the similarity of two senses $s_j$ and $s_k$ can be computed by the inner product of $Q_{\cdot,j}$ and $Q_{\cdot,k}$.

### 2.3 A Nuanced Sense Similarity: *wmfvec*

We can further use the features in WordNet to construct a better sense similarity measure. The most important feature of WN is senses are connected by relations such as *hypernymy, meronymy, similar attributes*, etc. We observe that neighbor senses are usually similar, hence they could be a good indicator for the latent semantics of the target sense.

We use WN neighbors in a way similar to *elesk*. Note that in *elesk* each definition is extended by including definitions of its neighbor senses. Also, they do not normalize the length. In our case, we also adopt these two ideas: (1) a sense is represented by the sum of its original latent vector and its neighbors' latent vectors. Let $N(j)$ be the set of neighbor senses of sense $j$. then new latent vector is: $Q_{\cdot,j}^{new} = Q_{\cdot,j} + \sum_k^{k \in N(j)} Q_{\cdot,k}$ (2) Inner product (instead of cosine similarity) of the two resulting sense vectors is treated as the sense pair similarity. We refer to our sense similarity measure as *wmfvec*.

[1] Due to limited space inference and update rules for $P$ and $Q$ are omitted, but can be found in (Srebro and Jaakkola, 2003)

## 3 Experiment Setting

**Task:** We choose the fine-grained All-Words Sense Disambiguation task, where systems are required to disambiguate all the content words (noun, adjective, adverb and verb) in documents. The data sets we use are all-words tasks in SENSEVAL2 [SE2], SENSEVAL3 [SE3], SEMEVAL-2007 [SE07], and Semcor. We tune the parameters in *wmfvec* and other baselines based on SE2, and then directly apply the tuned models on other three data sets.

**Data:** The sense inventory is WN3.0 for the four WSD data sets. WMF and LDA are built on the corpus of sense definitions of two dictionaries: WN and Wiktionary [Wik].[2] We do not link the senses across dictionaries, hence Wik is only used as augmented data for WMF to better learn the semantics of words. All data is tokenized, POS tagged (Toutanova et al., 2003) and lemmatized, resulting in 341,557 sense definitions and 3,563,649 words.

**WSD Algorithm:** To perform WSD we need two components: (1) a sense similarity measure that returns a similarity score given two senses; (2) a disambiguation algorithm that determines which senses to choose as final answers based on the sense pair similarity scores. We choose the Indegree algorithm used in (Sinha and Mihalcea, 2007; Guo and Diab, 2010) as our disambiguation algorithm. It is a graph-based algorithm, where nodes are senses, and edge weight equals to the sense pair similarity. The final answer is chosen as the sense with maximum indegree. Using the Indegree algorithm allows us to easily replace the sense similarity with *wmfvec*. In Indegree, two senses are connected if their words are within a local window. We use the optimal window size of 6 tested in (Sinha and Mihalcea, 2007; Guo and Diab, 2010).

**Baselines:** We compare with (1) *elesk*, the most widely used sense similarity. We use the implementation in (Pedersen et al., 2004).

We believe WMF is a better approach to model latent semantics than LDA, hence the second baseline (2) LDA using Gibbs sampling (Griffiths and Steyvers, 2004). However, we cannot directly use estimated topic distribution $P(z|d)$ to represent the definition since it only has non-zero values on one or two topics. Instead, we calculate the latent vector of a definition by summing up the $P(z|w)$ of all constituent words weighted by $X_{ij}$, which gives much better WSD results.[3] We produce LDA vectors [*ldavec*] in the same setting as *wmfvec*, which means it is trained on the same corpus, uses WN neighbors, and is tuned on SE2.

| Data | Model | Total | Noun | Adj | Adv | Verb |
|---|---|---|---|---|---|---|
| SE2 | random | 40.7 | 43.9 | 43.6 | 58.2 | 21.6 |
| | *elesk* | 56.0 | 63.5 | 63.9 | 62.1 | 30.8 |
| | *ldavec* | 58.6 | 68.6 | 60.2 | 66.1 | 33.2 |
| | *wmfvec* | **60.5** | **69.7** | **64.5** | **67.1** | **34.9** |
| | *jcn+elesk* | 60.1 | 69.3 | 63.9 | 62.8 | 37.1 |
| | *jcn+wmfvec* | **62.1** | **70.8** | **64.5** | **67.1** | **39.9** |
| SE3 | random | 33.5 | 39.9 | 44.1 | - | 33.5 |
| | *elesk* | 52.3 | 58.5 | 57.7 | - | 41.4 |
| | *ldavec* | 53.5 | 58.1 | 60.8 | - | 43.7 |
| | *wmfvec* | **55.8** | **61.5** | **64.4** | - | **43.9** |
| | *jcn+elesk* | 55.4 | 60.5 | 57.7 | - | 47.4 |
| | *jcn+wmfvec* | **57.4** | **61.2** | **64.4** | - | **48.8** |
| SE07 | random | 25.6 | 27.4 | - | - | 24.6 |
| | *elesk* | 42.2 | 47.2 | - | - | 39.5 |
| | *ldavec* | 43.7 | 49.7 | - | - | 40.5 |
| | *wmfvec* | **45.1** | **52.2** | - | - | **41.2** |
| | *jcn+elesk* | 44.5 | 52.8 | - | - | 40.0 |
| | *jcn+wmfvec* | **45.5** | **53.5** | - | - | **41.2** |
| Semcor | random | 35.26 | 40.13 | 50.02 | 58.90 | 20.08 |
| | *elesk* | 55.43 | 61.04 | 69.30 | 62.85 | 43.36 |
| | *ldavec* | 58.17 | 63.15 | 70.08 | **67.97** | 46.91 |
| | *wmfvec* | **59.10** | **64.64** | **71.44** | 67.05 | **47.52** |
| | *jcn+elesk* | 61.61 | 69.61 | 69.30 | 62.85 | 50.72 |
| | *jcn+wmfvec* | **63.05** | **70.64** | **71.45** | **67.05** | **51.72** |

Table 2: WSD results per POS ($K = 100$)

At last, we compare *wmfvec* with a mature WSD system based on sense similarities, (3) (Sinha and Mihalcea, 2007) [*jcn+elesk*], where they evaluate six sense similarities, select the best of them and combine them into one system. Specifically, in their implementation they use *jcn* for noun-noun and verb-verb pairs, and *elesk* for other pairs. (Sinha and Mihalcea, 2007) used to be the state-of-the-art system on SE2 and SE3.

## 4 Experiment Results

The disambiguation results ($K = 100$) are summarized in Table 2. We also present in Table 3 results using other values of dimensions $K$ for *wmfvec* and *ldavec*. There are very few words that are not covered due to failure of lemmatization or POS tag mismatches, thereby F-measure is reported.

Based on SE2, *wmfvec*'s parameters are tuned as $\lambda = 20, w_m = 0.01$; *ldavec*'s parameters are tuned as $\alpha = 0.05, \beta = 0.05$. We run WMF on WN+Wik for 30 iterations, and LDA for 2000 iterations. For

[2] http://en.wiktionary.org/

[3] It should be noted that this renders LDA a very challenging baseline to outperform.

LDA, more robust $P(w|z)$ is generated by averaging over the last 10 sampling iterations. We also set a threshold to *elesk* similarity values, which yields better performance. Same as (Sinha and Mihalcea, 2007), values of *elesk* larger than 240 are set to 1, and the rest are mapped to [0,1].

***elesk* vs *wmfvec***: *wmfvec* outperforms *elesk* consistently in all POS cases (noun, adjective, adverb and verb) on four datasets by a large margin (2.9% – 4.5% in $total$ case). Observing the results yielded per POS, we find a large improvement comes from nouns. Same trend has been reported in other distributional methods based on word co-occurrence (Cai et al., 2007; Li et al., 2010; Guo and Diab, 2011). More interestingly, *wmfvec* also improves verbs accuracy significantly.

***ldavec* vs *wmfvec***: *ldavec* also performs very well, again proving the superiority of latent semantics over surface words matching. However, *wmfvec* also outperforms *ldavec* in every POS case except Semcor adverbs (at least $+1\%$ in $total$ case). We observe the trend is consistent in Table 3 where different dimensions are used for *ldavec* and *wmfvec*. These results show that given the same text data, WMF outperforms LDA on modeling latent semantics of senses by exploiting missing words.

***jcn+elesk* vs *jcn+wmfvec***: *jcn+elesk* is a very mature WSD system that takes advantage of the great performance of *jcn* on noun-noun and verb-verb pairs. Although *wmfvec* does much better than *elesk*, using *wmfvec* solely is sometimes outperformed by *jcn+elesk* on nouns and verbs. Therefore to beat *jcn+elesk*, we replace the *elesk* in *jcn+elesk* with *wmfvec* (hence *jcn+wmfvec*). Similar to (Sinha and Mihalcea, 2007), we normalize *wmfvec* similarity such that values greater than 400 are set to 1, and the rest values are mapped to [0,1]. We choose the value 400 based on the WSD performance on tuning set SE2. As expected, the resulting *jcn+wmfvec* can further improve *jcn+elesk* for all cases. Moreover, *jcn+wmfvec* produces similar results to state-of-the-art unsupervised systems on SE02, 61.92% F-mearure in (Guo and Diab, 2010) using WN1.7.1, and SE03, 57.4% in (Agirre and Soroa, 2009) using WN1.7. It shows *wmfvec* is robust that it not only performs very well individually, but also can be easily incorporated with existing evidence as represented using *jcn*.

| dim | SE2 | SE3 | SE07 | Semcor |
|---|---|---|---|---|
| 50 | 57.4 - **60.5** | 52.9 - 54.9 | 43.1 - 44.2 | 57.90 - 58.99 |
| 75 | 57.8 - 60.3 | 53.5 - 55.2 | 43.3 - 44.6 | 58.12 - 59.07 |
| 100 | 58.6 - **60.5** | 53.5 - **55.8** | 43.7 - 45.1 | 58.17 - 59.10 |
| 125 | 58.2 - 60.2 | 53.9 - 55.5 | 43.7 - 45.1 | 58.26 - **59.19** |
| 150 | 58.2 - 59.8 | 53.6 - 54.6 | 44.4 - **45.9** | 58.13 - 59.15 |

Table 3: *ldavec* and *wmfvec* (latter) results per # of dimensions

### 4.1 Discussion

We look closely into WSD results to obtain an intuitive feel for what is captured by *wmfvec*. For example, the target word *mouse* in the context: ... *in experiments with mice that a gene called p53 could transform normal cells into cancerous ones... elesk* returns the wrong sense *computer device*, due to the sparsity of overlapping words between definitions of *animal mouse* and the context words. *wmfvec* chooses the correct sense *animal mouse*, by recognizing the biology element of *animal mouse* and related context words *gene, cell, cancerous*.

## 5 Related Work

Sense similarity measures have been the core component in many unsupervised WSD systems and lexical semantics research/applications. To date, *elesk* is the most popular such measure (McCarthy et al., 2004; Mihalcea, 2005; Brody et al., 2006). Sometimes people use *jcn* to obtain similarity of noun-noun and verb-verb pairs (Sinha and Mihalcea, 2007; Guo and Diab, 2010). Our similarity measure *wmfvec* exploits the same information (sense definitions) *elesk* and *ldavec* use, and outperforms them significantly on four standardized data sets. To our best knowledge, we are the first to construct a sense similarity by latent semantics of sense definitions.

## 6 Conclusions

We construct a sense similarity *wmfvec* from the latent semantics of sense definitions. Experiment results show *wmfvec* significantly outperforms previous definition-based similarity measures and LDA vectors on four all-words WSD data sets.

## Acknowledgments

This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the U.S. Army Research Lab. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

## References

Eneko Agirre and Aitor Soroa. 2009. Proceedings of personalizing pagerank for word sense disambiguation. In *the 12th Conference of the European Chapter of the ACL*.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

Samuel Brody, Roberto Navigli, and Mirella Lapata. 2006. Ensemble methods for unsupervised wsd. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*.

Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. 2007. Improving word sense disambiguation using topic features. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101.

Weiwei Guo and Mona Diab. 2010. Combining orthogonal monolingual and multilingual sources of evidence for all words wsd. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Weiwei Guo and Mona Diab. 2011. Semantic topic models: Combining word distributional statistics and dictionary definitions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Jay J. Jiang and David W. Conrath. 1997. Finding predominant word senses in untagged text. In *Proceedings of International Conference Research on Computational Linguistics*.

Linlin Li, Benjamin Roth, and Caroline Sporleder. 2010. Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*.

Rada Mihalcea. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the IEEE International Conference on Semantic Computing*, pages 363–369.

Nathan Srebro and Tommi Jaakkola. 2003. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning*.

Kristina Toutanova, Dan Klein, Christopher Manning, , and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.