

双方向LSTMによる分類語彙表番号を語義とした all-words WSD

著者	新納 浩幸, 鈴木 類, 古宮 嘉那子
雑誌名	言語資源活用ワークショップ発表論文集
巻	3
ページ	192-202
発行年	2018
URL	http://doi.org/10.15084/00001653

双方向 LSTM による分類語彙表番号を語義とした all-words WSD

新納浩幸 (茨城大学工学部情報工学科) *

鈴木類 (茨城大学大学院理工学研究科情報工学専攻) †

古宮嘉那子 (茨城大学工学部情報工学科) ‡

All-words WSD with WLSP number as a Sense Label Using a Bidirectional LSTM

Hiroyuki Shinnou (Ibaraki University, Department of Computer and Information Sciences)

Rui Suzuki (Ibaraki University, Major in Computer and Information Sciences)

Kanako Komiya (Ibaraki University, Department of Computer and Information Sciences)

要旨

語義曖昧性解消は意味解析の重要な要素技術であるが、実際のシステムに利用されることは少ない。これは現状の語義曖昧性解消が主として教師あり学習のアプローチをとっているため、対象単語が限定されてしまうからである。我々は対象単語を限定しない all-words WSD システムを、大規模な語義タグ付きコーパスと双方向 LSTM を用いて構築した。構築できたシステムは、入力されたテキスト内の全ての単語にその語義を高精度に付与できる。本論文では構築したシステムを紹介し、その有用性を文書分類タスクと語義のクラスタリングから示す。

1. はじめに

語義曖昧性解消は意味解析の根幹の処理でありながら、そのシステムが現実のアプリケーションで広く利用されているとは言いがたい。これは現状の語義曖昧性解消が、主として、教師あり学習のアプローチをとっているため、対象単語が限定されてしまうことが大きな原因である。そのため対象単語を限定せず、入力文内の単語全てにその語義を付与する all-words WSD (Word Sense Disambiguation) の重要性が古くから指摘されてきた (Navigli (2009))。しかし全ての多義語に対して語義タグ付きの用例データベースを構築するコストは膨大であるため、単純に通常の WSD の手法を拡張するだけでは実現できない。辞書の用例文などを用いた知識ベースの手法や教師なし学習の手法を用いる試みもあるが、多くの場合、MFS (Most Frequent Sense) と同等以下の精度しか得られず、実用的な精度とは言えない (Kulkarni et al. (2010))。

一方、語義タグ付きコーパスの開発は並行して行われており、昨年、国立国語研究所から BCCWJ に対する分類語彙表番号アノテーションデータ (Kato et al. (2017)) が公開された。

* hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

† 17nm709g@vc.ibaraki.ac.jp

‡ kanako.komiya.nlp@vc.ibaraki.ac.jp

これを語義タグ付きの用例データベースと見なせば, all-words WSD は実現できるが, 全ての多義語に対する用例データベースとしては小規模であり, 個々の多義語の WSD を解くという直接的な手法は利用できない. ここでは all-words WSD を各単語に語義 (ラベル) を与える系列ラベリング問題としてみなし, 双方向 LSTM (Long-ShortTerm Memory) を用いて all-words WSD システムを構築した.

本論文では構築したシステムの詳細を述べる. また, その有用性を示すために2つの実験を行う. 1つは文書分類のタスクである. 通常, 文書分類は文書を bag of words のモデルを使って素性ベクトルに変換するが, 本システムを利用して素性ベクトルに語義の素性を加えることができる. 語義の素性を追加することで識別精度が向上する.

もう1つは語義のクラスタリングである. 本システムを利用してコーパスを語義列のコーパスに変換できる. さらに得られた語義列のコーパスを word2vec (Mikolov et al. (2013) Mikolov et al. (2013)) にかけることにより, 語義の分散表現が得られる. この語義の分散表現を利用して, 語義のクラスタリングを行う. 本システムの語義は分類語彙表番号であり, 分類語彙表が概念辞書であるため, 得られたクラスタリング結果と分類語彙表を比較でき, 主観的に構築されている分類語彙表の考察が行える.

2. 関連研究

all-words WSD はドメインを限定すれば通常の教師あり学習も可能である. 実際に SemEval で行われた all-words WSD のタスクでも, いくつかの教師あり学習によるシステムが参加している. ただし教師あり学習はその拡張性に問題がある.

教師あり学習手法を用いない場合, all-words WSD の手法は知識ベースの手法か教師なし学習手法に分類できる (Kulkarni et al. (2010)). 知識ベースの手法として, 古典的には Lesk の手法 (Lesk (1986)) がある. これは対象単語の周辺の単語集合と, 対象単語の各語義の定義文中に現れる単語集合との重なり度合いを調べ, その度合いの大きい語義を選択するというものである. ただし一般に知識ベースの手法は語義の頻度の情報を利用していないために, 精度が低いという問題がある.

教師なし学習手法には様々なタイプのもものが存在するが (Yarowsky (1995), Izquierdo-Beviá et al. (2006), Zhong and Ng (2009)), 近年は, 語義列の生成モデルを定義し, ある種のヒューリスティックを導入することでプレーンなコーパスから生成モデルのパラメータを推定する手法が採られている (Boyd-Graber et al. (2007), Tanigaki et al. (2013, 2015), Komiya et al. (2015)). 教師なし学習手法は知識に基づく手法よりも精度は高いが, 実用的な精度とは言えない. さらに教師なし学習手法による all-words WSD システムは, 通常の WSD システムとは異なる入出力となっていることにも注意すべきである. 通常の WSD システムの入力は WSD の対象単語を含む文であり, 出力はその対象単語の語義である. 一方, 教師なし学習手法による all-words WSD システムの入力はコーパスである. 入力コーパス中のすべての単語に語義を付与する. しかし新たに対象単語を含む文を単独で入力しても, その対象単語に語義を付与することはできない.

また近年は語義の分散表現を求めることで all-words WSD を実現することも試みられてい

る (Neelakantan et al. (2014), Chen et al. (2014)). 単語 w に対する i 番目の語義の分散表現 s_i とする. w の周辺文脈を分散表現 v で表現し, s_i と v の類似度を測り, 最も類似度の高い i に対する語義を識別結果とすることで, WSD が行える. この手法は知識ベースの手法の一種であり, この手法も精度が低いという問題がある. 一般に MFS よりも高い精度は得られない. そのため逆に分散表現を用いて, MFS を推定する方法も研究されている (Bhingardive et al. (2015)).

一方, Hatori は all-words WSD を通常の WSD の対象単語を拡張する形ではなく, all-words WSD を系列ラベリング問題として見なして解いている (Hatori et al. (2008)). ただし系列ラベリング問題を解く手法として CRF を用いているために, 対象単語に対する語義付き用例が少ない場合に, 統一した枠組みでの処理ができない. このような問題に対処するために, Shinnou は CRF ではなく点推定を用いた日本語 all-words WSD システム KyWSD (Shinnou et al. (2017)) を開発したが, 単語切り自体もシステム内の学習で行っており, 一般の単語切りとは異なる単位を用いているので, その点で実用的ではない⁽¹⁾.

また本システムと同じく分類語彙表番号を語義とした日本語 all-words WSD の研究として (Suzuki et al. (2018)) がある. そこでは対象単語とその類義語の周辺単語の分散表現の距離を計算する教師なし学習のアプローチで語義曖昧性解消を行っている. この研究も教師なしであるため新たな入力文に対して語義を付与することができず, 実用的ではない.

以上の観点から本システムの特徴を述べると, 本システムは all-words WSD を系列ラベリング問題として見なして, 双方向 LSTM を用いて解いている. そのためその精度は高く, 語義のラベル数が多い場合や, ある単語について用例が少ない場合でも, 同一の枠組みで処理が行える. また教師なし学習ではないために, 新たな入力文に対しても語義を付与することができる.

3. 双方向 LSTM による系列ラベリング

all-words WSD は系列ラベリング問題とみなすことができる. 系列ラベリング問題をニューラルネットワークで扱う場合, リカレントニューラルネットワーク (以下, RNN) を使用する. RNN では時刻 t の中間層の内容を時刻 $t-1$ の入力に使い状態を保持しながら学習することで時系列に対応しているが, 長い系列データを学習する際に勾配の消失が起こる場合がある. LSTM は RNN の一種種であり, 入力, 出力, 忘却の 3 つのゲート処理を取り入れることで勾配の消失を防ぎ長い系列データにも対応することができる. また RNN は時系列データを扱うものであり, 自然言語処理では文や文書の単語列を時系列データとみなして使っている. そのため通常, 注目している時刻 t 以降の単語も利用できるため, データを逆方向からも解析できる. 順方向の LSTM と逆方向の LSTM を同時に用いて, 時刻 t での出力を求めるのが双方向 LSTM である (図 1 参照).

モデルで利用した訓練データは BCCWJ に対する分類語彙表番号アノテーションデータである. このデータは 340,879 トークン, 19,432 タイプ (記号空白を含む), 918 種類の語義で構

⁽¹⁾ 用言の語幹と語尾を分離する形で単語切りが行われている.

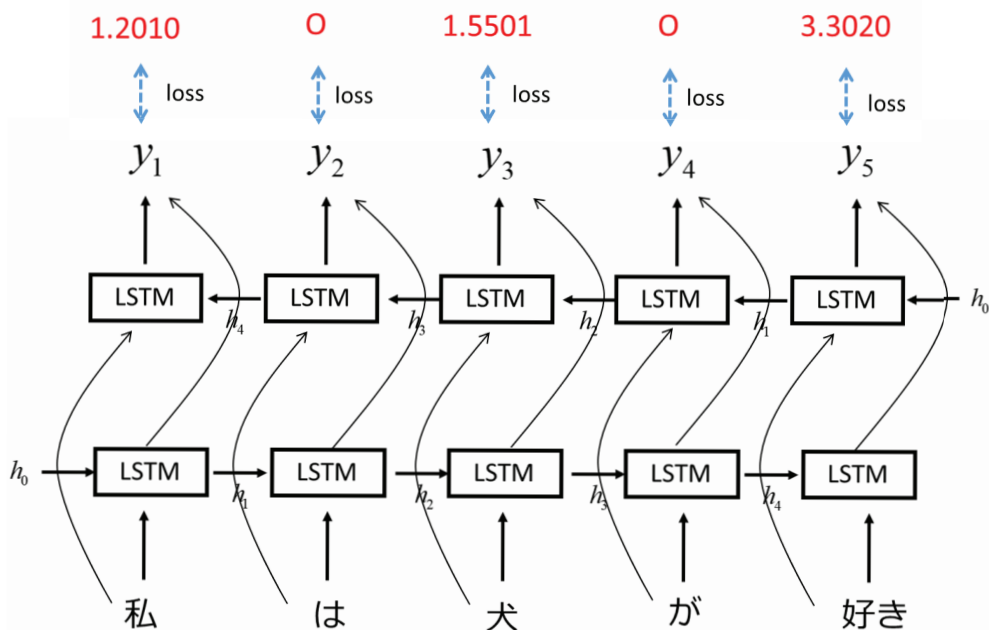


図1 双方向 LSTM による語義の付与

成されている。このデータのうち1割をテストデータのために取り除いておき、残りの9割で双方向 LSTM の学習を行った。LSTM は2階層を用い、単語から分散表現に変換する部分は学習を行わずに既存の日本語分散表現データである nwjc2vec (Shinnou et al. (2017)) を用いた。epoch は20回で終了させた。

4. システム概要

本研究で作成したシステムは先に構築した双方向 LSTM のモデルを核とするが、形態素解析や語義の候補の算出部分は別の資源を利用している。これらを組み合わせて本システムの all-words WSD が実現できている。システムへの入力は一平なテキストであり、出力はそのテキストが形態素解析され自立語に語義（分類語彙表番号）が付与されたものである。分類語彙表番号を持たない助詞などの単語には語義として‘O’というラベルを付与している。本システムの実行例を以下に示す。

システムの入力から出力までのステップは以下のとおりである。

step.1 単語切りと品詞付け

単語切りには、辞書として unidic-cwj-2.3.0 を用いた形態素解析器 MeCab を使用した。

step.2 語義候補の取り出し

各単語の語義候補の取り出しには国語研より公開されている分類語彙表番号-Unidic 語彙表番号対応表 wls2unidic を用いた。wls2unidic は分類語彙表番号から Unidic 語彙表番号を求めることも、Unidic 語彙表番号から分類語彙表番号を求めることもできる。本研究では分類語彙表番号を語義としているため、MeCab で出力される各単語の語彙表番号をもとに

```

> cat sample.txt
学问に基づいた本物の技術を創出できるのはやはり大学だと実感し始めていたからである。

> ./allwordswsd sample.txt
学问-名詞/1.3074 に-助詞/0 基づい-動詞/2.1110 た-助動詞/0 本物-名詞/1.1040
の-助詞/0 技術-名詞/1.3421 を-助詞/0 創出-名詞/1.3200 できる-動詞/2.1220
の-助詞/1.1000 は-助詞/0 やはり-副詞/4.3120 大学-名詞/1.2630 だ-助動詞/0
と-助詞/0 実感-名詞/1.3001 し-動詞/2.3430 始め-動詞/2.1502 て-助詞/0
い-動詞/2.1200 た-助動詞/0 から-助詞/0 で-助動詞/0 ある-動詞/2.1200
. -補助記号/0

```

図2 all-word WSD の実行例

wlsp2unidic を用いて語義候補となる分類番号を取り出した。

step.3 分散表現への変換

各単語は分散表現に変換されて LSTM へ入力される。分散表現は LSTM の学習時に同時学習することもできるが、ここでは既存の分散表現 nwjc2vec (Shinnou et al. (2017)) を利用した。

step.4 LSTM による語義の確率算出

システムを構築する際の訓練データは 918 種類の語義で構成されている。したがって LSTM によって出力されるのは入力したテキストの各単語に対する 919 個の語義（語義なしを含む）の確率となる。

step.5 語義候補との調整と出力

システムが出力するのは、入力テキストの各単語に語義を付与したものである。**step.2** で語義候補が 1 つだった場合はその語義を付与し、語義候補が複数あった場合はその中で最も確率が高いものを **step.4** から求め付与する。

5. 精度測定の実験

システムの精度測定の実験について述べる。本実験では、分類語彙表番号アノテーションデータの 9 割を訓練データとして双方向 LSTM で学習し、1 割をテストデータとした。テストデータには 1,736 種類の多義語が含まれており、多義語の総数は 7,137 である。また、多義語の総語義数は 19,320 で平均語義数は 2.71 である。ただし、ここでは Section 4 の **step.5** の出力の調整は行わず、LSTM の出力の中で最も確率の高い語義を付与している。実験の結果、すべての単語の正解率は 0.870 (0.768)、多義語の正解率は 0.737 となった。多義語の平

均語義数が 2.71 なので、語義をランダムで選択した場合の正解率は 0.369 である。MFS によるすべての単語の正解率は 0.827 であった。このため理論的に本システムの多義語の正解率は MFS 以上の値であることから、本システムの精度は高いと考えられる。

また、ここで求めた全単語の正解率の内訳は、単義語の正解率 0.786、多義語の正解率 0.737 であるが、これは LSTM の出力である。つまり全語義の中から最も確率の高いものを出力していることに注意したい。実際のシステムが語義を出力する際は、対象単語の候補の語義のうち最も確率の高いものを出力すればよい。この点を考慮に入れた場合には正解率は更に高くなると予想できる。

6. 文書分類への応用

本章では本論文で構築した all-words WSD システムの応用として文書分類を行う。

文書分類は文書を bag of words のモデルを用いて素性ベクトルに変換し、その上で SVM などの機械学習手法を利用して解決する。bag of words のモデルでは、通常、素性として自立語を用いるが、all-words WSD システムを利用することで、その自立語の語義も素性として利用できる。

実験データは以下で公開されている Amazon Dataset の日本語文書を利用した。

<https://www.uni-weimar.de/en/media/chairs/computer-science-department/webis/data/corpus-webis-cls-10/#webis-download>

上記データは感情分析データであり、3つの領域 ('books (B)', 'DVD (D)' 及び 'music (M)') からなっている。また各領域に訓練データとテストデータがそれぞれ 2,000 文書存在する。つまり計 12,000 の文書が存在する。この 12,000 文書に対して形態素解析を行い、自立語を素性として、各文書を素性ベクトルで表し、Naive Bayes により識別した (表 1 の BOW)。また上記 12,000 文書に対して all-words WSD を行い、自立語の他に自立語の語義も素性として加えて、各文書を素性ベクトルで表し、Naive Bayes により識別した (表 1 の BOW+sense)。実験の結果を表 1 に示す。

Domain	BOW	BOW+sense
(B) books	0.6820	0.6765
(D) DVD	0.7265	0.7420
(M) music	0.7645	0.7725
Average	0.7243	0.7303

表 1 文書分類の実験結果

更に領域適応の実験も行った。先でのデータでは 3つの領域 (B,D,M) があるので、領域シフトのその組み合わせとして 6通りがある。それぞれの領域シフトにおいてソース領域の訓練データにより Naive Bayes の分類器を作成し、それをターゲット領域のテストデータの正解率

で評価した。ソース領域の訓練データでは素性として自立語のみのも (BOW) と自立語の他に自立語の語義を加えてたもの (BOW+sense) を各々試し、比較する。結果を表 2 に示す。

DA	BOW	BOW+sense
B → D	0.6775	0.6825
B → M	0.6500	0.6610
D → B	0.6620	0.6715
D → M	0.6975	0.6945
M → B	0.6460	0.6455
M → D	0.6910	0.6925
Average	0.6707	0.6746

表 2 文書分類の領域適応

以上の実験から文書分類では自立語だけではなくその語義も素性として加える効果が確認できた。

7. 語義のクラスタリングへの応用

本章では本論文で構築した all-words WSD システムの応用として、分類語彙表番号の分散表現を構築し、語義のクラスタリングを行う。

7.1 語義の分散表現の構築

語義のクラスタリングを行うために、まず語義の分散表現を構築した。手順は以下のとおりである。

最初に毎日新聞 '93 年から '99 年の記事からランダムに 30 万文を取り出し、これを分散表現構築のためのコーパスとした。次に本システムを用いてこのコーパス中の全単語に分類語彙表番号を付与した。次にシステムの出力から分類語彙表番号の分かち書きを作成した。このとき語義が O (つまり自立語ではない) の場合、つまりその単語が自立語ではない場合は、語義ではなく単語表記とした。

分類語彙表番号の分かち書きは 7,937,723 トークン、12,698 タイプから構成されている。このように分かち書きされたテキストに対して word2vec (Mikolov et al. (2013) Mikolov et al. (2013)) を用いて分類語彙表番号の分散表現を構築した。次元数は 100 とした。用いた word2vec の学習のパラメータを表 3 に示す。

7.2 語義のクラスタリング

前章で構築した語義の分散表現のうち、名詞に対応する語義だけを対象にしてクラスタリングを行った。対象とした語義は 509 種類である。利用したクラスタリング手法は Word 法である。得られたデンドログラムを図 4 に示す。

次にこのクラスタリング結果を評価する。分類語彙表は概念辞書であるために、語義が階層構造 (木構造) の中で定義されている。そのため同一親ノードまでの木の深さにより、擬似的

学問-名詞/1.3074 に-助詞/O 基づい-動詞/2.1110 た-助動詞/O 本物-名詞/1.1040
 の-助詞/O 技術-名詞/1.3421 を-助詞/O 創出-名詞/1.3200 できる-動詞/2.1220
 の-助詞/1.1000 は-助詞/O やはり-副詞/4.3120 大学-名詞/1.2630 だ-助動詞/O
 と-助詞/O 実感-名詞/1.3001 し-動詞/2.3430 始め-動詞/2.1502 て-助詞/O
 い-動詞/2.1200 た-助動詞/O から-助詞/O で-助動詞/O ある-動詞/2.1200
 . -補助記号/O



1.3074 に 2.1110 た 1.1040 の 1.3850 を 1.3200 2.1220 1.1000 は 4.3120
 1.2630 だと 1.3001 2.3430 2.1502 て 2.1200 た から で 2.1200 .

図3 語義の分かち書き

CBOW or skip-gram	-cbow	1
Dimensionality	-size	100
# of surrounding words	-window	5
# of negative samples	-negative	5
Hierarchical softmax	-hs	0
Mini. sample threshold	-sample	1e-3
# of iterations	-iter	5

表3 word2vec のパラメータ値

に語義間の距離を測ることができる。具体的には語義番号の桁数の位置が木の階層を表し、その桁数の数値がその階層でのクラスを表している。例えば語義番号‘1.1344’と‘1.1310’では‘1.13**’の部分が共通の親ノードに対応するため、距離は3となる。‘1.1611’と‘1.1612’では‘1.161*’の部分が共通の親ノードとに対応するため、距離は2となる。このように語義間に距離を定義すると、クラスタ内の距離が定義できる。クラスタ内の距離とはクラスタ内の各要素の間の距離の平均である。このクラスタ内の距離の平均をクラスタリング結果の評価値と見なせる。

先にクラスタリングして得られたデンドログラムを‘cut’することで、50個のクラスタに分割した。各クラスタについてクラスタ内の距離を調べ、その平均を取ると3.054となった。クラスタリング対象となった509個の語義の集合からランダムに10個選びそれを1つのクラスタと考えて、クラスタ内の距離を調べる実験を20回行い、その平均を取ると3.716であった。このことから本論文で得た語義のクラスタリング結果は妥当性があると考えられる。

また参考としてクラスタ内の距離が最も小さかったクラスタは以下である。その距離は

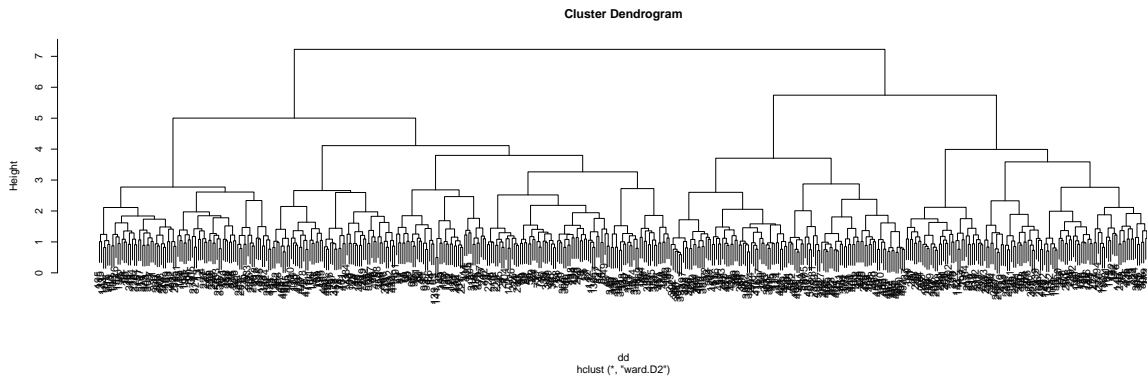


図4 語義のクラスタリング結果

1.733 であった。

1.3101, 1.3105, 1.3110, 1.3111, 1.3112, 1.3150

またクラスタ内の距離が最も大きかったクラスタは以下である。その距離は 3.821 であった。

1.1100, 1.1623, 1.2030, 1.3047, 1.3300, 1.3360, 1.4580, 1.5000

これらの点から分類語彙表は概念の階層構造を考察できる。

8. おわりに

本論文では分類語彙表番号を語義とした all-words WSD システムを作成した。訓練データには BCCWJ に対する分類語彙表番号アノテーションデータを使用し、学習には双方向 LSTM を用いた。構築したシステムは高い識別精度を示した。また構築したシステムの応用として文書分類と語義のクラスタリングを行うことで、本システムの有用性を示した。

謝 辞

本研究（の一部）は国立国語研究所コーパス開発センターの共同研究プロジェクト「コーパスアノテーションの拡張・統合・自動化に関する基礎研究」（2016-2021 年度）の成果である。

文 献

Roberto Navigli (2009). “Word sense disambiguation: A survey.” *ACM Computing Surveys (CSUR)*, 41:2, p. 10.

Anup Kulkarni, Mitesh M Khapra, Saurabh Sohoney, and Pushpak Bhattacharyya (2010).

- “CFILT: Resource conscious approaches for all-words domain specific WSD.” *SemEval-2010*, pp. 421–426.
- Sachi Kato, Masayuki Asahara, and Makoto Yamazaki (2017). “Annotation of ‘Word List by Semantic Principles’ information on ‘Balanced Corpus of Contemporary Written Japanese’.” *Processing of NLP 2017*, pp. 306–309 (In Japanese).
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Efficient Estimation of Word Representations in Vector Space.” *ICLR Workshop paper*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). “Distributed representations of words and phrases and their compositionality.” *Advances in neural information processing systems*, pp. 3111–3119.
- Michael Lesk (1986). “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.” *the 5th annual international conference on Systems documentation*, pp. 24–26.
- David Yarowsky (1995). “Unsupervised word sense disambiguation rivaling supervised methods.” *ACL-95*, pp. 189–196.
- Rubén Izquierdo-Beviá, Lorenza Moreno-Monteagudo, Borja Navarro, and Armando Suárez (2006). “Spanish all-words semantic class disambiguation using Cast3LB corpus.” *MICAI 2006: Advances in Artificial Intelligence*, pp. 879–888.
- Zhi Zhong, and Hwee Tou Ng (2009). “Word Sense Disambiguation for All Words without Hard Labor.” *IJCAI-2009*, pp. 1616–1622.
- Jordan L Boyd-Graber, David M Blei, and Xiaojin Zhu (2007). “A Topic Model for Word Sense Disambiguation.” *EMNLP-CoNLL-2007*, pp. 1024–1033.
- Koichi Tanigaki, Mitsuteru Shiba, Tatsuji Munaka, and Yoshinori Sagisaka (2013). “Density Maximization in Context-Sense Metric Space for All-words WSD.” *ACL-2013*, pp. 884–893.
- Koichi Tanigaki, Shuichi Tokumoto, Tatsuji Munaka, and Yoshinori Sagisaka (2015). “Hierarchical Bayesian word sense disambiguation for mapping context space to sense space (in Japanese).” *IPSJ SIG on NLP*, pp. NL-220–5.
- Kanako Komiya, Yuto Sasaki, Hajime Morita, Hiroyuki Shinnou, Minoru Sasaki, and Yoshiyuki Kotani (2015). “Surrounding Word Sense Model for Japanese All-words Word Sense Disambiguation.” *PACLIC-29*, pp. 35–43.
- Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum (2014). “Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space.” *EMNLP-2014*, pp. 1059–1069.
- Xinxiong Chen, Zhiyuan Liu, and Maosong Sun (2014). “A Unified Model for Word Sense Representation and Disambiguation.” *EMNLP-2014*, pp. 1025–1035.
- Sudha Bhingardive, Dharendra Singh, Rudramurthy V, Hanumant Harichandra Redkar, and Pushpak Bhattacharyya (2015). “Unsupervised Most Frequent Sense Detection us-

- ing Word Embeddings.” *HLT-NAACL 2015*, pp. 1238–1243.
- Jun Hatori, Yusuke Miyao, and Jun’ichi Tsujii (2008). “Word Sense Disambiguation for All Words using Tree-Structured Conditional Random Fields.” *COLING-2008*, pp. 43–46.
- Hiroyuki Shinnou, Kanako Komiya, Minoru Sasaki, and Shinsuke Mori (2017). “Japanese all-words WSD system using the Kyoto Text Analysis ToolKit.” *PACLIC-31*, No.11.
- Rui Suzuki, Kanako Komiya, Masayuki Asahara, Minoru Sasaki, and Hiroyuki Shinnou (2018). “All-words Word Sense Disambiguation Using Concept Embeddings.” *LREC-2018*.
- Hiroyuki Shinnou, Masayuki Asahara, and Minoru Sasaki Kanako Komiya (2017). “nwjc2vec: Word Embedding Data Constructed from NINJAL Web Japanese Corpus (In Japanese).” *Natural Language Processing*, 24:5, pp. 705–720.