CMPE 409 Machine Translation

Word Sense Disambiguation

Murat ORHUN
Adapted from the original slides of Raymond J. Mooney

Istanbul Bilgi University

June 1, 2022

Word-sense disambiguation (WSD) is an open problem in computational linguistics concerned with identifying which sense of a word is used in a sentence.
The solution to this issue impacts other computer-related writing, such as discourse, improving relevance of search engines, anaphora resolution, coherence, and inference.

Lexical Ambiguity

- Most words in natural languages have multiple possible meanings.
  - "pen" (noun)
    - The dog is in the pen.
    - The ink is in the pen.
  - "take" (verb)
    - Take one pill every morning.
    - Take the first right past the stoplight.
- Syntax helps distinguish meanings for different parts of speech of an ambiguous word.
  - "conduct" (noun or verb)
    - John's conduct in class is unacceptable.
    - John will conduct the orchestra on Thursday.

Word Sense Disambiguation (WSD)

- Many tasks in natural language processing require disambiguation of ambiguous words.
  - Question Answering
  - Information Retrieval
  - Machine Translation
  - Text Mining
  - Phone Help Systems
- Understanding how people disambiguate words is an interesting problem that can provide insight in psycholinguistics.

Sense Inventory

- What is a “sense” of a word?
  - Homonyms (disconnected meanings)
    - bank: financial institution
    - bank: sloping land next to a river
  - Polysemes (related meanings with joint etymology)
    - bank: financial institution as corporation
    - bank: a building housing such an institution
- Sources of sense inventories
  - Dictionaries
  - Lexical databases

WordNet

- A detailed database of semantic relationships between English words.
- Developed by famous cognitive psychologist George Miller and a team at Princeton University.
- About 144,000 English words.
- Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called synsets.

WordNet Synset Relationships

- Antonym: front → back
- Attribute: benevolence → good (noun to adjective)
- Pertainym: alphabetical → alphabet (adjective to noun)
- Similar: unquestioning → absolute
- Cause: kill → die
- Entailment: breathe → inhale
- Holonym: chapter → text (part to whole)
- Meronym: computer → cpu (whole to part)
- Hyponym: plant → tree (specialization)
- Hypernym: apple → fruit (generalization)

EuroWordNet

- EuroWordNet is a system of semantic networks for European languages, based on WordNet.[1]
- Each language develops its own wordnet but they are interconnected with interlingual links stored in the Interlingual Index (ILI).
- Unlike the original Princeton WordNet, most of the other wordnets are not freely available.
- https://en.wikipedia.org/wiki/EuroWordNet

EuroWordNet

- Dutch
- Italian
- Spanish
- German
- French
- Czech
- Estonian
- These wordnets are now frozen, but wordnets for other languages have been developed to varying degrees.

EuroWordNet data samples as Binary Databases

ReadMe.txt	Download and read first!!!!!!!!!!!
Periscope.zip (1.63 MB)	Files for installing Periscope: graphical interface to the EuroWordNet database.
Manuals.zip (0.43 MB)	Manuals describing the Top Ontology and the Periscope Viewer.
ili.zip (20.4MB)	The Inter-Lingual-Index for matching synsets across wordnets.
ewnDatabaseWithWordNet15.zip (62 KB)	Main database files for all wordnets samples, including WordNet1.5
ewnDatabaseWithoutWordNet15.zip (61 KB)	Main database files for all wordnets samples, without WordNet1.5
ewnSamples.zip (2.78 MB)	Sample databases for Dutch, Spanish, Italian, French, German, Czech, Estonian and the EuroWordNet top-ontology.
wn15.zip (52.7 MB)	WordNet1.5 as EuroWordNet database.

https://archive.illc.uva.nl//EuroWordNet/data/sampleData.html

EuroWordNet data samples as ASCII files in EuroWordNet

import.zip (0.23 MB)	Dutch, Italian, Spanish, French, German, Czech and Estonian Samples and the EuroWordNet top-ontology as Ascii import files.
wnNimport.zip (6,41 MB)	The nominal synsets of WordNet1.5 in the EuroWordNet import format, separated into 3 parts.
wnVimport.zip (1,81 MB)	The verbal synsets of WordNet1.5 in the EuroWordNet import format
wnAimport.zip (0.99MB)	The adjectival synsets of WordNet1.5 in the EuroWordNet import format
wnRimport.zip (0.17 MB)	The adverbial synsets of WordNet1.5 in the EuroWordNet import format

https://archive.illc.uva.nl//EuroWordNet/data/sampleData.html

WordNet Senses

- WordNets senses (like many dictionary senses) tend to be very fine-grained.
- "play" as a verb has 35 senses, including
  - play a role or part: "Gielgud played Hamlet"
  - pretend to have certain qualities or state of mind: "John played dead."
- Difficult to disambiguate to this level for people and computers. Only expert lexicographers are perhaps able to reliably differentiate senses.
- Not clear such fine-grained senses are useful for NLP.
- Several proposals for grouping senses into coarser, easier to identify senses (e.g. homonyms only).

Senses Based on Needs of Translation

- Only distinguish senses that are translate to different words in some other language.
    - play: tocar vs. jugar
    - know: conocer vs. saber
    - be: ser vs. estar
    - leave: salir vs dejar
    - take: llevar vs. tomar vs. sacar
- May still require overly fine-grained senses
    - river in French is either:
        - fleuve: flows into the ocean
        - rivière: does not flow into the ocean

Learning for WSD

- Assume part-of-speech (POS), e.g. noun, verb, adjective, for the target word is determined.
- Treat as a classification problem with the appropriate potential senses for the target word given its POS as the categories.
- Encode context using a set of features to be used for disambiguation.
- Train a classifier on labeled data encoded using these features.
- Use the trained classifier to disambiguate future instances of the target word given their contextual features.

Feature Engineering

- The success of machine learning requires instances to be represented using an effective set of features that are correlated with the categories of interest.
- Feature engineering can be a laborious process that requires substantial human expertise and knowledge of the domain.
- In NLP it is common to extract many (even thousands of) potentially features and use a learning algorithm that works well with many relevant and irrelevant features.

Contextual Features

- Surrounding bag of words.
- POS of neighboring words
- Local collocations
- Syntactic relations

Experimental evaluations indicate that all of these features are useful; and the best results comes from integrating all of these cues in the disambiguation process.

Surrounding Bag of Words

- Unordered individual words near the ambiguous word.
- Words in the same sentence.
- May include words in the previous sentence or surrounding paragraph.
- Gives general topical cues of the context.
- May use feature selection to determine a smaller set of words that help discriminate possible senses.
- May just remove common "stop words" such as articles, prepositions, etc.

POS of Neighboring Words

- Use part-of-speech of immediately neighboring words.
- Provides evidence of local syntactic context.
- P_{-i} is the POS of the word *i* positions to the left of the target word.
- P_i is the POS of the word *i* positions to the right of the target word.
- Typical to include features for:

  $P_{-3}, P_{-2}, P_{-1}, P_1, P_2, P_3$

Local Collocations

- Specific lexical context immediately adjacent to the word.
- For example, to determine if "interest" as a noun refers to "readiness to give attention" or "money paid for the use of money", the following collocations are useful:
  - "in the interest of"
  - "an interest in"
  - "interest rate"
  - "accrued interest"
- $C_{i,j}$ is a feature of the sequence of words from local position i to j relative to the target word.
  - $C_{-2,1}$ for "in the interest of" is "in the of"
- Typical to include:
  - Single word context: $C_{-1,-1}$, $C_{1,1}$, $C_{-2,-2}$, $C_{2,2}$
  - Two word context: $C_{-2,-1}$, $C_{-1,1}$, $C_{1,2}$
  - Three word context: $C_{-3,-1}$, $C_{-2,1}$, $C_{-1,2}$, $C_{1,3}$

Syntactic Relations (Ambiguous Verbs)

- For an ambiguous verb, it is very useful to know its direct object.
  - "played the game"
  - "played the guitar"
  - "played the risky and long-lasting card game"
  - "played the beautiful and expensive guitar"
  - "played the big brass tuba at the football game"
  - "played the game listening to the drums and the tubas"
- May also be useful to know its subject:
  - "The game was played while the band played."
  - "The game that included a drum and a tuba was played on Friday."

Syntactic Relations (Ambiguous Nouns)

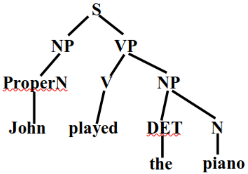
- For an ambiguous noun, it is useful to know what verb it is an object of:
  - "played the piano and the horn"
  - "wounded by the rhinoceros' horn"
- May also be useful to know what verb it is the subject of:
  - "the bank near the river loaned him $100"
  - "the bank is eroding and the bank has given the city the money to repair it"

Syntactic Relations (Ambiguous Adjectives)

- For an ambiguous adjective, it useful to know the noun it is modifying.
  - "a brilliant young man"
  - "a brilliant yellow light"
  - "a wooden writing desk"
  - "a wooden acting performance"

Using Syntax in WSD

- Produce a parse tree for a sentence using a syntactic parser.



- For ambiguous verbs, use the head word of its direct object and of its subject as features.
- For ambiguous nouns, use verbs for which it is the object and the subject as features.
- For ambiguous adjectives, use the head word (noun) of its NP as a feature.

Approaches

There are four conventional approaches to WSD:

- Dictionary- and knowledge-based methods: These rely primarily on dictionaries, thesauri, and lexical knowledge bases, without using any corpus evidence.
- Semi-supervised or minimally supervised methods: These make use of a secondary source of knowledge such as a small annotated corpus as seed data in a bootstrapping process, or a word-aligned bilingual corpus.
- Supervised methods: These make use of sense-annotated corpora to train from.
- Unsupervised methods: These eschew (almost) completely external information and work directly from raw unannotated corpora. These methods are also known under the name of word sense discrimination.

Dictionary- and knowledge-based methods

- The Lesk algorithm[20] is the seminal dictionary-based method.
- It is based on the hypothesis that words used together in text are related to each other and that the relation can be observed in the definitions of the words and their senses.
- Two (or more) words are disambiguated by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions.
- For example, when disambiguating the words in "pine cone", the definitions of the appropriate senses both include the words evergreen and tree (at least in one dictionary).

Lesk algorithm

- The Lesk algorithm is a classical algorithm for word sense disambiguation introduced by Michael E. Lesk in 1986.
- The Lesk algorithm is based on the assumption that words in a given "neighborhood" (section of text) will tend to share a common topic
- A simplified version of the Lesk algorithm is to compare the dictionary definition of an ambiguous word with the terms contained in its neighborhood.

Lesk algorithm

Versions have been adapted to use WordNet.[2] An implementation might look like this:

- for every sense of the word being disambiguated one should count the amount of words that are in both neighborhood of that word and in the dictionary definition of that sense
- the sense that is to be chosen is the sense which has the biggest number of this count

Lesk algorithm

A frequently used example illustrating this algorithm is for the context "pine cone". The following dictionary definitions are used:

```
PINE
1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness
```

```
CONE
1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of certain evergreen trees
```

https://en.wikipedia.org/wiki/Lesk_algorithm

Further Reading

- Jurafsky, D. and J. H. Martin. Speech and language processing*:* An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition, Upper Saddle River, NJ: Prentice-Hall, 2008. (chapter 20).