# Word Sense Disambiguation: Pipelining with Supervised LSA

Kiran Vodrahalli, Evelyn Ding, Albert Lee

May 8, 2014

## 1 Introduction

### 1.1 Problem Statement

Multiple word senses

ex. bank: "financial institution" or "river bank" or "count on something happening" or "supply/stock held in supply for future use" ...

Wrong sense can drastically alter meaning

Objective: given context, determine which sense of the word is intended

First proposed by Weaver in 1949 as computational task in the early days of MT

One of the oldest open problems in computational linguistics

AI-Complete Problem

Example

"Paintings, drawings and sculpture from every period of art during the last 350 years will be on display ranging from a Tudor portrait to contemporary British art." vs. "Through casual meetings at cafe's, the artists drew together to form a movement in protest against the waste of war, against nationalism and against everything pompous, conventional or boring in the art of the Western world."

the creation of beautiful or significant things vs. the products of human creativity; works of art collectively

## 2 Literature Survey

knowledge based

unsupervised corpus-based

supervised corpus-based

### 2.1 Knowledge-based

Lesk's algorithm: two words W1 and W2 with multiple word senses for each sense i in W1 for each sense j in W2 calculate overlap of i and j in dictionary

definitions maximize overlap i and j to determine word senses Semantic similarity: calculate path between words in wordnet for each word sense with other words in the context minimize total semantic path to select word sense context can vary from one sentence to length of document

## 2.2 Unsupervised Corpus-based

Token-based discrimination: cluster contexts in which a given target word appears Cluster 1: The line was occupied. The operator came into the line abruptly.

Cluster 2: The line was really long and it took forever to be served. I stood in the queue for about 10 minutes

Use parallel corpus across language to infer sense distinctions across the language Je vais prendre ma propre dcision. vs. Je vais prendre ma propre voiture.

## 2.3 Supervised Corpus-based

Use annotated corpus to build model ex. to know: 1) be aware of piece of information I want to know who is winning. I know that the president lied. 2) know person She doesn't know the composer. Do you know my sister? Use context of how word usually appears (bigram/trigram models) if (feature) then word sense

## 2.4 LSA – Past Efforts

LSA was one of the first approaches to make use of the co-occurrence model for word meaning. The idea, as implemented in the famous 1998 paper by Landauer, is that given a set of corpuses, we build a term-document matrix where words are on the rows and documents are on the columns, and the frequencies are in the corresponding cells. Then we apply SVD to this matrix. This approach is essentially unsupervised because it relies on clustering (word co-occurrence matrix) to derive similarities between documents and words.

Originally, the LSA approach was primarily intended to remove uninformative singular values, and then reconstruct the term-document matrix (Landauer et al 1998). The idea there was to compare documents to see how similar they were to each other with noise removed. Part of their motivation was to see if co-occurrence representations was in fact how humans represented meaning themselves.

(Katz et al) first applied LSA to WSD in an unsupervised approach by folding new documents with ambiguous word into the semantic space created by the SVD of the term-document matrix, and then used cosine-based clustering. We do a supervised implementation of this approach.

It is worth noting that other similar approaches exist in recent years that are probably more powerful (but more difficult to implement). We detail them in the next slide.

– Probabilistic LSA: Formulated as early as 1999, this approach is based on mixture decomposition from latent class model. (this is a latent variable model for co-occurrence data) Model fitting is done with Expectation Maximization (EM).

– Non-Negative Matrix Factorization: An alternative to using SVD is NNMF, which is a matrix factoring technique introduced by Lee and Seung (2000). Van de Cruys (2011) makes use of this technique: – minimize Kullback-Leibler divergence instead of Euclidean distance

– minimizing Kullback-Leibler divergence is better, since minimizing Euclidean distance makes the normal distribution assumption for language – language is not Gaussian. – useful for extraction of semantic dimension – turns out to be useful to have no negative relations encoded (all matrix values are positive)

Van de Cruys (2011): uses this approach on SEMEVAL-2010 dataset. The rest of the approach is analogous to SVD-LSA, computing the centroid of candidate vectors and comparing via Kullback-Leibler divergence to the target vector to determine word sense.

# 3 Data

Senseval-2 (2001) dataset 73 words that need to be disambiguated ex. fine, art, pull, turn, carry Up to 43 senses per word 8611 labeled training instances 4328 test instances

brown corpus

# 4 Algorithms

Part of speech tagging and bigram models Use for subset of data that has high predictability Supervised LSA for the rest of the data

Other methods had low accuracy

## 4.1 Part-of-Speech

## 4.2 Semantic Knowledgebase

## 4.3 Supervised LSA

Our approach uses simple LSA with a supervised learning component.

1) Build term-document matrix a) terms are words in Brown Corpus and in training data b) documents are categories of Brown, and concatenated paragraphs of training data with same word sense for ambiguous word 2) Apply SVD to get $U$, $D$, $V^T$.

Supervised part: 3) For each (paragraph, ambiguous word, word sense) tuple in the training data: a) create vector in the term-document space b) fold this vector into the reduced semantic space of V c) use cosine similarity to come up

with the eigentopic most associated with the paragraph. Then, we have built an association between each word sense and an eigentopic vector in V. 4) For each (paragraph, ambiguous word) in testing data: a) come up with the eigentopic vector associated b) For that eigentopic vector, there is a probability for each word sense for a given word – we pick the highest probability word sense for the word we are disambiguating

In other words, for a given eigentopic, we will always pick the same word sense for a given word. This approach has its flaws.

We also tried a slightly different approach to capture more of the information: For each word, and for each sense of the word, we created a vector by applying cosine similarity to each eigentopic column vector for our training vector (created from paragraph). We averaged over all the training data for each word sense. Then for testing, we created this vector from the test paragraph and compared via cosine similarity again to determine which word sense was most likely. This approach had issues and we got very bad results. It might be better in the future to try a modified approach to this (i.e. do something other than average the vectors).

# 5 Results

From Senseval-2:

Best overall (supervised approach): 64%

Best unsupervised approach: 40%

## 5.1 POS Bigram Model

POS Bigram Model (w/frequency count ¿ 3)

Threshold 80%40%0% Correctly identified 199518574 Total identified 2538931136 % Accuracy 78.7%58.0%50.5% % Coverage 5.8%20.6%26.2%

POS Bigram Model (w/frequency count ¿ 0)

Threshold 80%40%0% Correctly identified 466808850 Total identified 89816411826 % Accuracy 51.9%49.2%46.5% % Coverage 20.7%38.7%42.2%

Correctly disambiguates: "Marketing too, in its strictest sense, is outside our remit. " (the way in which a word or expression or situation can be interpreted)

"Before he lost his money he had the good sense to commission John Soane to design the Campden Hill Square house." (a general conscious awareness)

Supervised LSA: a) With only the Brown Corpus categories as column vectors (15 column vectors -¿ reduced to 9) in the term-document matrix: 42% b) With the Brown Corpus categories + the word-sense categories (in total, 861 column vectors -¿ reduced to 140 column vectors after SVD): 38% – highly dependent on the training data – different results for different words, some words had lower percentages, some had higher percentages (as low as 7%, as high as 61%) – easier to do ambiguous words with fewer senses

# 6 Conclusions

One issue with the LSA approach may be that the corpuses we use (namely, for a given word and a given sense, all the training paragraphs concatenated together) may actually cover a very broad array of topics, and thus a lot of noise might be added. While this should theoretically be removed by SVD, it might just be that the 'paragraph topics' were too disjoint and the correlation for different very fine-tuned usage is not strong enough. Examples of failures:

for the word "fine":

our program: "elegant" actual: "thin (in thickness)" our program: "superlative" actual: "elegant"

and so on... it makes sense that both senses could apply to some of the testing data

Labeled training data sparsity is a large issue

Some statistical methods fall short of disambiguating subtle nuances between word senses

The LSA approach is primarily topic categorization – we need to use better documents that are more tailored to WSD, perhaps.

It is hard to use a topic-categorization approach to do fine-tuned WSD.

# 7 Future Work

Could try different document columns for Supervised LSA. Could try applying supervised approach to PLSA or NNMF-LSA.

Recall Socher et al. (2012) – Recursive Neural Nets on sentiment trees Integrate with Socher's work on sentiment analysis Found that Socher doesn't disambiguate across certain words such as "mean", "like" Could improve accuracy of sentiment analysis

Can be applied to translation if WSD is made better

# 8 Citations

Need to get list of papers and cite them in text as well. These are on the Google Doc and also in Kiran's Chrome tabs. Also we all have the book Albert sent out.