BOOTSTRAPPING IN WORD SENSE DISAMBIGUATION

DOINA TĂTAR

Abstract. The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word: Allen (1995), Manning, Schutze (1999), Tătar (2005). In this paper we present one algorithm (Tătar, Şerban 2001) which combines Yarowsky's principles (Yarowsky 1999) (one sense per discourse and one sense per collocation) and the Naive Bayes Classifier method.

1. INTRODUCTION

The word sense disambiguation (WSD) is probably one of the most important open problem and it has now already a long "history" in computational linguistics. WSD is the process of identifying the correct meanings of words in particular contexts. More exactly, the problem that arise in a natural language is that many words present the phenomenon of polysemy that means they have several meanings or senses. That sense depends on what context they occur. The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word. Whenever a system's actions depend on the meaning of the text being processed, WSD is necessary. WSD has direct applications in some fields of text understanding as information retrieval, text summarization, machine translation (Orăşan *et al.* 2003). It is only an intermediate task in Natural Language Processing (NLP), like POS tagging or parsing.

The algorithms used in WSD are classified to whether they involve supervised or unsupervised learning. Unsupervised learning can be viewed as clustering task while supervised learning is usually seen as a classification task. Dictionary based disambiguation, can be considered as intermediary between supervised and unsupervised disambiguation. The algorithm presented in this paper is a bootstrapping one, that means it repeatedly convert training corpus and test corpus, during some specific actions. We also remember in this paper the k-NN algorithm, as used in one international on-line contest on WSD (Şerban, Tătar 2004).

Notational conventions used in the following are (Manning, Schutze 1999, Tătar, Şerban 2001, Tătar 2003):

w-- the word to be disambiguated (target word);
s1, … ,sN1 --- possible senses for w;
c1, … ,cN2 --- contexts of w in corpus;
v1, … ,vN3 --- words used as contextual features for disambiguation of w.

RRL, **LI**, *3–4*, p. 415–420, Bucureşti, 2006

BDD-A241 © 2006 Editura Academiei
Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)

Concerning of v1,… vN3, there are two possibilities: they are collocates or co-occurrences with w. In the first case the contextual features occur in a fixed position near w, in a window centered on w. In the second case, the contextual features occur together with w, in arbitrarily positions. Large debates are initiated regarding which kind of features are better; the conclusion drawn was that only the aspect of corpus and the goal of studies are dictating the selection.

Yarowsky (1999) observed that there are constraints between different occurrences of contextual features that can be used for disambiguation. Two such constraints are formulated as principles:

- Principle "One sense per discourse": the sense of a target word is highly consistent within a given discourse (document) ;
- Principle "One sense per collocation": the contextual features (nearby words) provide strong clues to the sense of a target word.

For example, regarding the first constraint, for the word plant, if his sense is in a first occurrence "living being", then later occurrences are likely to refer to "living beings" too. As the second principle, if the word animal occurs together with plant, this word is likely to be a clue word for the "living beings" sense.

2. SUPERVISED DISAMBIGUATION BY NAIVE BAYES CLASSIFIER ALGORITHM

A supervised disambiguation requires a tagged corpus with semantic senses. That means, each occurrence of an ambiguous word w is annotated with its sense. These tagged with senses corpora are usually made by human expert. Such annotated corpus was constructed and used in on-line contest Senseval (Şerban and Tătar 2004). The task for Naive Bayes Classifier is to build a tool which correctly classifies a new context based on the contextual features occurring in this context.
What a Naive Bayes Classifier (NBC) realizes is the calculus of the sense s' which, for the target word w and a given context c, satisfies the relation:
$s' = \text{argmax}_{sk} P(sk \mid c) = \text{argmax}_{sk} P(c \mid sk) / P(c) \, P(sk) = \text{argmax}_{sk} P(c \mid sk) P(sk)$

Here P(sk | c) represents the probability of the sense sk conditioned by the context c and it is rewritten by the Bayes rule in (P(c | sk) /P(c)) P(sk).
As probability P(c) does not depend on sk , it can be omitted from calculus of argmax.

The same value for s' is obtained if we consider the logarithm of expression:
$s' = \text{argmax}_{sk}(\log P(c \mid sk) + \log P(sk))$

The "Naïve" Bayes assumption is that the contextual features are all conditional independent. That means :
$P(c \mid sk) = P(\{ vj \mid vj \text{ is in } c \} \mid sk) = \prod_{vj} P(vj \mid sk)$

Thus the probability P(c | s k) is simply the product of the probabilities P (v j | s k) for all features vj. This naive assumption has two consequences:

BDD-A241 © 2006 Editura Academiei
Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)

- the structure and order of words in context is ignored;
- the presence of one word in the context doesn't depend on the presence of another.

This is clearly not true. Though, there is a large number of cases in which the algorithm works well. In Manning, Shutze (1999) is reported that a disambiguation system based on this algorithm is correct for about 90 percents of cases.

As regarding the probabilities P(vj | sk) and P(sk), these are calculated from the labeled (annotated) corpus with the formulas: P(vj | sk)=C(vj,sk)/C(sk) and P(sk)= C(sk)/C(w). Here C(vj, sk) is the number of occurrences of vj in all the contexts annotated with the sense sk, C(sk) is the number of all contexts with the sense sk and C(w) is the total number of occurrences of the word w.

3. A BOOTSTRAPPING ALGORITHM ON THE BASE OF THE PRINCIPLES: ONE SENSE PER DISCOURSE AND ONE SENSE PER COLLOCATION, USING NBC

The algorithm begins by identification for a small number of training contexts. This could be accomplished by hand tagging with senses the contexts of w for which the sense of w is clear because some seed collocations occur in these contexts. This annotation is made on the base of the dictionaries or by using WordNet (for English). The initial set of annotated contexts is used for learning an naive bayesian classifier NBC. This NBC will help in annotating new contexts. By repeating the process, the annotated part of corpus grows. We will stop when the remaining not annotated corpus is empty or any new context can be annotated, if a stop condition is established.

The notational conventions used in following are as above. Let us consider that the set of words V={v1,…,vl} is a small subset of the set of features v1, … ,vN3. They are firmly associated with the senses for w, such that the occurrence of vi in the context of w determines the choice for w of a sense, according to the principle of one sense per collocation. For example, for the word *plant*, the occurrence in the same context of the word *life* or *animal* means a sense (let say A), while the occurrence in the same context of the word *industrial* or *construction* means another sense (let say B).

Thus, some contexts from the following set of contexts obtained as query results with corpus Cobuild can be solved, and they will be associated with the sense A or B.

- For A:

Context:

aspect, features and animal and *plant* life." [p] [p] These were never

- For B:

Contexts:

be far less than buying a brand new *plant*. [p] A Phillipines industrial

BDD-A241 © 2006 Editura Academiei
Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)

the planning and construction of the *plant* at Rabta near Tripoli and were
industrial equipment and engineering *plant*. [p] The company insures

Some others contexts are not yet annotated with sense, as for example:

hard currency. And so we've found a plant, and I have some seeds here from
a chunk was hacked off the mother plant and deposited in my car. [p] Mary's
coal deliveries to one preparation plant and occupied the offices of Peabody
down and dig out the roots, but don't plant another cherry in the same spot as
A. japonica Japanese aucuba. A male plant, bearing panicles of purple-
satisfies me. I've finally arrived at plant combinations that provide three
in strata with specially designed plant-growing pockets, the finished
by layering low whippy shoots [p] Plant heathers in soil fortified with
and experience of any individual plant in my garden alone is hardly
40.000 from a van taking them from a plant in Staines, Surrey, used to store
Basrah, and other cities. A four-acre plant in Baghdad's western outskirts was
Day of 1937 at the Republic Steel plant in Chicago, where ten people were
3–4in. deep down the row and plant into that, then the soil is
cement inside steel drums. The first plant is now in successful operation. [p]
particular pose (something at which Plant is particularly adept), taking
Basket and Container displays. [p] Plant just one variety by itself to make
aspect, features and animal and plant life." [p] [p] These were never

At the next step, calculate from the set of annotated contexts the new probabilities P(sk), P(vj | sk), where sk is A or B. Some context cnew will be annotated with the sense s', where s' is $s'=\text{argmax}_{sk}$ P(sk | cnew). This has as a result the modification of the set V of features, as a set of words with maximum frequency in the new annotated set of contexts, for each sense. Let mention that for a sense of word w we can have a set of features. If v in V is a word with a maxime frequency which occurs in the context c solved with the sense sj, then v is a feature for sense sj, according with the principle “one sense per discourse”.

The algorithm:

Let us denote by Cann the set of contexts aready adnotated by senses (as in our example with *plant*) and by Cnotann the rest of contexts. The algorithm consists in the following structure:

While Cnotann is not Φ do
For each new contex cnew from Cnotann
For each sense sk, feature vj
Calculate P(sk), P(vj | sk)
Endfor
Calculate $s'=\text{argmax}_{sk}$ P(sk | cnew)
Cann= Cann U {cnew}
Cnotann= Cnotann \ {cnew}

BDD-A241 © 2006 Editura Academiei
Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)

Modify the features set of vj
Endfor
Endwhile
Senseval3 contest.

In 2004 a system developed at Babeş-Bolyai University (Şerban, Tătar 2004) participated at the third context in evaluating sense, Senseval3, contest developed in Marts-April 2004. Our system falls in the supervised learning approach category: it is trained to learn a classifier that can be used to assign a yet unseen example to one of a fixed number of senses. We received from the organizers of contest a trained corpus (a number of annotated contexts), where the system learned the classifier, and a test corpus which the system annotated. In our system we use the Vector Space Model: a context is represented as a vector c of some features. This vector of a context of the target word w is defined as:

c =(w1, …, wN3) where wi is the weight of th feature vi. The similarity between two contexts is the normalized cosine between the vectors associated with the two contexts (Jurafsky, Martin 2000, Manning, Schutze 1999).

The weight wi could be the frequency of the feature vi (term frequency), denoted by fi. Another method to establish the weight wi is to capture the fashion of distribution of vi in all the set of contexts by principle: "Features that are limited to a small number of contexts are useful for discriminating those contexts. Features that occur frequently across the entire set of contexts are less useful in this discrimination". In this case we use a new weight for a feature, called "inverse document frequency", denoted by idf and defined as bellow:

Let us consider that the number of contexts is N2 and the number of contexts in which the feature vi occurs is ni. The inverse document frequency of the feature vi is calculated as:

$$idfi = N2/ni \quad or \quad idfi=\log(N2/ni)$$

Combining the tf with idf we obtain tf.idf weighting. In this case:

c =(w1, …, wN3) , where $wi=fi \cdot idfi$.

The algorithm used by our system at Senseval3 contest (Şerban, Tătar 2004) is that of k-NN or memory based learning (Manning, Schutze 1999, Şerban, Tătar 2004). At training time, our k-NN model memorizes all the contexts in the training set by their associated features. Later, when proceeds a new context cnew, the classifier first selects k contexts in the training set that are closest to cnew, then pick the best sense for cnew.

k-NN algorithm

TRAINING Calculate vector c for each context;

BDD-A241 © 2006 Editura Academiei
Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)

TEST Calculate the set:

$$A= \{ c \mid sim(cnew, c) \text{ is maxim}, |A| = k \}$$

that means A is the set of the k nearest neighbors contexts of cnew.

Calculate :

$$Score(cnew, sj) = \Sigma_{ci \text{ in } A} (sim(cnew, ci) \cdot aij)$$

where aij is 1 if vector ci has the sense sj and aij is 0 otherwise.

Finally, calculate $s' = argmax_{sj} Score(cnew, sj)$.

The precision of our system was much more higher than of a baseline algorithm, which assert to each sense of a word the most common sense in the set of contexts. We intend to improve our system and to combine elements of k-NN methods with elements of our bootstrapping algorithm, presented in this paper.

REFERENCES

Allen J., 1995, *Natural language understanding*, (2nd ed.), Menlo Park, Benjamin/Cummings.

Jurafsky, D., J. Martin, 2000, *Speech and language processing*, Prentice Hall.

Manning, C., H. Schutze, 1999, *Foundation of statistical natural language processing*, MIT Press.

Orăşan, C., D. Tătar, G. Şerban, D. Avram, A. Oneţ, 2003, "How to build a QA system in your back-garden: application to Romanian", EACL '03, Budapest, 139-142.

Resnik, P., D. Yarowsky, 1998, "Distinguishing Systems and Distinguishing sense: new evaluation methods for WSD", *Natural Language Engineering*, 1, nr 1.

Şerban, G., D. Tătar, 2004, "UBB system at Senseval3", *Proceedings of Workshop in Word Disambiguation*, ACL 2004, Barcelona, July 2004, 226-229.

Tătar, D., G. Şerban, 2001, "A new algorithm for WSD", *Studia Univ. "Babeş-Bolyai", Informatica*, 2, 99-108.

Tătar, D., 2005, "Word Sense Disambiguation by machine learning", *Fundamenta Informaticae*, IOS Press, 64, 1-4, 433-442.

Tătar, D., 2003, *Inteligenţa artificială: aplicaţii în prelucrarea limbajului natural*, Editura Microinformatică.

Yarowsky, D., 1999, *Hierarchical Decision Lists for WSD*, Kluwer Academic Publishers.

BDD-A241 © 2006 Editura Academiei

Provided by Diacronia.ro for IP 216.73.216.121 (2026-02-25 00:54:23 UTC)