T-61.5020 Statistical Natural Language Processing

Answers 7 — Word sense disabiguation
Version 1.0

1. Let's start from Bayes' theorem.

$$P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)}$$

Now we are interested only in the order of probabilities, not the absolute values, so we can forget the normalization term $P(c)$:

$$\begin{aligned} s' &= \operatorname*{argmax}_{s_k} \frac{P(c|s_k)P(s_k)}{P(c)} \\ &= \operatorname*{argmax}_{s_k} P(c|s_k)P(s_k) \end{aligned}$$

In the equation the latter term is the prior for the word sense. It can be estimated for example by calculating how many of the words in the training corpus have appeared in the sense s_k. For now we concentrate on the term $P(c|s_k)$.

Let's choose the nearest 10 words as the context:

$$c = (\ w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\ )$$

The word we are studying would be in the middle, $w_{4.5}$. Here, the order of the context words makes a difference, and this is marked by using the parentheses. Using these kind of feature vectors is in practice impossible, as two equal 10 word contexts are not likely to be found in the training or test sets. We approximate the model by assuming that the order is not significant (now using brackets):

$$c = \{\ w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\ \}$$

Now we have:

$$P(c|s_k) = P(\{\ w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\ \}|s_k)$$

Let's make the estimation easier by assuming that the words occur independently:

$$\begin{aligned} & P(\{\ w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\ \}|s_k) \\ = & P(w_0|s_k)P(w_1|s_k)\ldots P(w_9|s_k) \\ = & \prod_{i=0}^{9} P(w_i|s_k) \end{aligned}$$

Finally, let's write the expression open:

$$
\begin{aligned}
s' &= \operatorname*{argmax}_{s_k} P(c|s_k)P(s_k) \\
&= \operatorname*{argmax}_{s_k} \left(P(s_k) \prod_{i=0}^{9} P(w_i|s_k) \right) \\
&= \operatorname*{argmax}_{s_k} \left(\log P(s_k) + \sum_{i=0}^{9} \log P(w_i|s_k) \right)
\end{aligned}
$$

In the last row the formula is written in logarithmic form. This can be done, because taking the logarithm does not affect the order of the values.

None of the used approximations is totally correct, but the roughest error is probably the one of independency of the context words. However, this is the way of getting an easily feasible method.

2. Let's use the formula derived in the previous problem:

$$
\begin{aligned}
s' &= \operatorname*{argmax}_{s_k} P(c|s_k)P(s_k) \\
&= \operatorname*{argmax}_{s_k} \left(P(s_k) \prod_{i=0}^{N} P(w_i|s_k) \right),
\end{aligned}
$$

where w_i are the words that occurred in the context.

We need two estimates: probability $P(w_j|s_k)$ that the word w_j in the context occurs with the sense s_k, and prior probability $P(s_k)$. As we have equal number of occurrences for the senses *sataa=rain* and *sataa=number*, we can but set the prior to 0.5.

Maximum likelihood (ML) estimation is applied in the course book. In our problem we were asked to use priors, so let's define a small prior that all words are of equal probability to the probability $P(w_j|s_k)$, and add it to the estimators with coefficient $\lambda = 0.5$. This can be thought as if every known word had already occurred 0.5 times in both context types. A large λ emphasises the meaning of the prior, and thus a small evidence from the training set does not change it much.

$$
P(w_j|s_k) = \frac{C(w_j, s_k) + \lambda}{C(s_k) + N\lambda}
$$

N is the number of known words, 85.

a) Let's calculate the estimators needed in the first test sentence:

$$\begin{aligned}
P(\text{“koirasusitarha”}|\text{”sataa”=rain}) &= \frac{0.5}{6+0.5\cdot 85} = \frac{1}{97} \\
P(\text{“vieraili”}|\text{”sataa”=rain}) &= \frac{1}{97} \\
P(\text{“pari”}|\text{”sataa”=rain}) &= \frac{1}{97} \\
P(\text{“ihminen”}|\text{”sataa”=rain}) &= \frac{1}{97}
\end{aligned}$$

We see that for the first sense, all probability mass comes from the prior. For the comparison number (i.e. unnormalized probability) we get

$$0.5 \cdot \frac{1}{97} \cdot \frac{1}{97} \cdot \frac{1}{97} \cdot \frac{1}{97} = 5.6 \cdot 10^{-9}$$

The same calculations for the number sense:

$$\begin{aligned}
P(\text{“koirasusitarha”}|\text{”sataa”=number}) &= \frac{0.5}{6+0.5\cdot 85} = \frac{1}{97} \\
P(\text{“vieraili”}|\text{”sataa”=number}) &= \frac{1}{97} \\
P(\text{“pari”}|\text{”sataa”=number}) &= \frac{2+0.5}{6+0.5\cdot 85} = \frac{5}{97} \\
P(\text{“ihminen”}|\text{”sataa”=number}) &= \frac{5}{97}
\end{aligned}$$

The comparison number is:

$$0.5 \cdot \frac{1}{97} \cdot \frac{1}{97} \cdot \frac{5}{97} \cdot \frac{5}{97} = 1.4 \cdot 10^{-7}$$

So according to the model, the number sense of “sataa” is more probable.

b) As we saw before, we can leave out all the words that have not occurred in the contexts of either word, as they do not affect the order of the comparison numbers. Let's use the tool that changes all disambigous numbers to string “num”. The needed probabilities are:

$$
\begin{aligned}
P(\text{“räntää”}|\text{”sataa”=rain}) &= \frac{1.5}{6+0.5\cdot 85} = \frac{3}{97} \\
P(\text{“tai”}|\text{”sataa”=rain}) &= \frac{3}{97} \\
P(\text{“lunta”}|\text{”sataa”=rain}) &= \frac{7}{97} \\
P(\text{“num”}|\text{”sataa”=rain}) &= \frac{5}{97}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{“räntää”}|\text{”sataa”=number}) &= \frac{0.5}{6+0.5\cdot 85} = \frac{1}{97} \\
P(\text{“tai”}|\text{”sataa”=number}) &= \frac{1}{97} \\
P(\text{“lunta”}|\text{”sataa”=number}) &= \frac{1}{97} \\
P(\text{“num”}|\text{”sataa”=number}) &= \frac{5}{97}
\end{aligned}
$$

For the comparison numbers we get

$$
\begin{aligned}
\text{sataa=rain} &: 0.5\cdot\frac{3}{97}\cdot\frac{3}{97}\cdot\frac{7}{97}\cdot\frac{3}{97} = 1.1\cdot 10^{-6} \\
\text{sataa=number} &: 0.5\cdot\frac{1}{97}\cdot\frac{1}{97}\cdot\frac{1}{97}\cdot\frac{5}{97} = 2.8\cdot 10^{-8}
\end{aligned}
$$

This time the word seems to mean raining.

c) For the third sentence,

$$
\begin{aligned}
P(\text{“noin”}|\text{”sataa”=rain}) &= \frac{3}{97} \\
P(\text{“num”}|\text{”sataa”=rain}) &= \frac{5}{97}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{“noin”}|\text{”sataa”=number}) &= \frac{7}{97} \\
P(\text{“num”}|\text{”sataa”=number}) &= \frac{5}{97}
\end{aligned}
$$

For the comparison numbers we get

$$
\begin{aligned}
\text{sataa=rain} &: 0.5\cdot\frac{3}{97}\cdot\frac{5}{97} = 8.5\cdot 10^{-8} \\
\text{sataa=number} &: 0.5\cdot\frac{7}{97}\cdot\frac{5}{97} = 2.0\cdot 10^{-7}
\end{aligned}
$$

So it seems to be a number here.

d) For the last sentence the given training data does not change the probabilities for any direction. And because the priors were equal, the model cannot make any decision here.

3. Let's find the dictionary definitions for the words in the tested sentence. Those are compared to the dictionary definitions of two senses of the studied word. The meaning that has more mutual words with the words in the dictionary definitions of the other words (including the word itself) in the sentence is decided to be the correct one.

   In this case, from the definition of "ampuminen", shooting, we find the words "harjoitella" and "varusmies" that are also in the test sentence. The word "sarjatuli" is found from the definition of "kivääri", so three points for shooting.

   From the definition of "ammuminen", that is moo'ing, we find the word "niityllä", which is also in the test sentence. One point for moo'ing.

   It seems that it is shooting for this one $(3 > 1)$.

4. Let's see how many hits the Google will give:

prices	go up	111 000
price	goes up	88 100
		199 100

prices	slant	58
prices	lean	2 520
prices	lurch	21
price	slants	1
price	leans	63
price	lurches	114
		2 777

   This example goes clearly for the sense *"go up"*.

   What about the next example? If we do the translation and search using the given word order, we will get no hits (excluding the hits for this exercise problem). So we try to find documents where the words may occur in any order:

   We see that the verb meanings of the words win here, altough the nouns would probable be more correct. All searches are not even needed, because the first one already produces more hits than all of the other senses together. In addition, most of the hits returned by the first four searches were from dictionaries.

want	shin	hoof	liver	or	snout	260
like	shin	hoof	liver	or	snout	304
covet	shin	hoof	liver	or	snout	219
desire	shin	hoof	liver	or	snout	243
						1 026
want	kick	poke	cost	or	suffer	43 500

As the senses *shin, hoof, liver* and *snout* are much rarer than the verbs, they are found much less. In this situation we should probably normalize the search in some way. This example was harder than the first one also because this time the sentence was not a common and fixed phrase.

5. The problem is to estimate the probability of the sense s_k when we know the context c_i.

$$P(s_k|c_i) = \frac{P(c_i|s_k)P(s_k)}{\sum_{k'=1}^{K} P(c_i|s_{k'})P(s_{k'})}$$

Let's use the Naive Bayes assumption presented in first problem, i.e. that the words w_j in the context do not depend on each other:

$$P(c_i|s_k) = \prod_{w_j \in c_i} P(w_j|s_k)$$

Initialization

Let's initialize the parameters:

- Set all the words to be equally probable for both sources, and add some noise σ. Without the noise the algorithm will not converge, as all the events have equal probabilities.

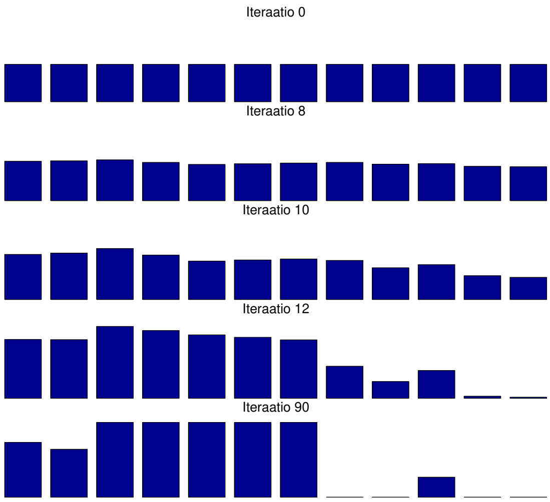
$$P(w_j|s_k) = \frac{1}{J} + \sigma$$

Here J is the number of the known words.

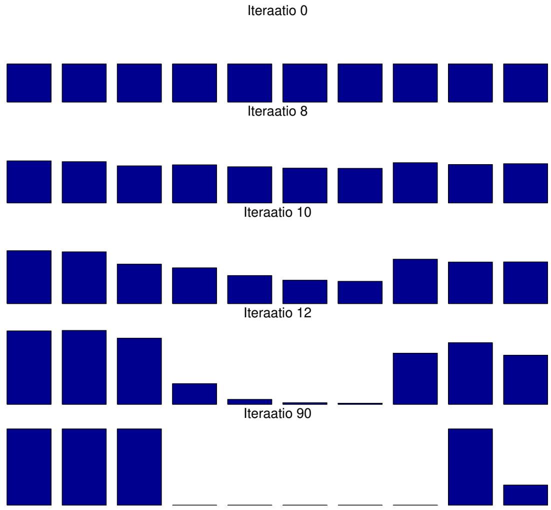
- Set all senses to be of equal probability.

$$P(s_k) = \frac{1}{K}$$

Here K is the number of the different senses.



(a) $P(w_j|s_0)$: Probability that the sense s_0 produces the word w_j



(b) $P(s_0|c_i)$: Probability that the context c_i appears in the sense s_0

Figure 1: *EM algorithm. The figure illustrates the convergence. Word from left to right: yksi, kaksi, kolme, neljä, viisi, seitsemän, kahdeksan, mänty, leppä, haapa, koivu, kataja. Sentences are in the same order as in the problem.*

E-step

– Calculate the probability of each sense for all contexts:

$$P(s_k|c_i) = \frac{\prod_{w_j \in c_i} P(w_j|s_k)P(s_k)}{\sum_{k'=1}^{K} \prod_{w_j \in c_i} P(w_j|s_{k'})P(s_{k'})}$$

M-step

– Estimate the new word probabilities using the sentence probabilities estimated in the E-step.

$$P(w_j|s_k) = \frac{\sum_{c_i : w_j \in c_i} P(s_k|c_i)}{\sum_{k'=1}^{K} \sum_{c_i : w_j \in c_i} P(s_{k'}|c_i)}$$

– Update the prior probabilities:

$$P(s_k) = \frac{\sum_{i=1}^{I} P(s_k|c_i)}{\sum_{k'=1}^{K} \sum_{i=1}^{I} P(s_{k'}|c_i)}$$

The convergence of the algorithm, as E- and M-steps are iterated, is illustrated in Figure 1. In this case the priors $P(s_k)$ we kept at $\frac{1}{2}$ for first 15 iterations, which improved the stability. We see that the algorithm can separate the numbers and the trees. For sentences 8 and 9 the model overlearns and sets them only to one sense.

If the amount of training data would be larger, also these estimates might be more feasible.

The same algorithm can be used to, e.g., separate a set of documents to their topics. In that case, the contexts would be the full documents.

6. Here we present one possible example solution step by step. The most important points where we have made an arbitary decision that can increase inaccuracy and could be as well done otherwise, are marked with *italics*.

   1) The first step is to clean all the extra headlines, tags and markings away. Then we want to separate the contexts. *Let the context for each word to be the full sentence where it occurs.* Let's change some two words, e.g. "sade" (rain) and "komissio" (commission), to a common pseudoword. At the same time we can collect the correct answers for evaluation purpose.
   2) Let's change all words of the contexts to a vector form. Here we could use binary indicator vectors, but let's *approximate those by setting a random 200-dimensional vector to each word*. If the vector has enough dimensions, it is roughly orthogonal between all different words and the approximation is quite good.
   3) Let's assume that *the order of the context words does not matter*. Let's calculate the context for each word by *summing up the vectors of the words in its context and dividing the sum by the number of the vectors*.
   4) Let's cluster the context vectors *using the self-organizing map (SOM)*. The number by clusters can be decided *experimentally*. For a small number of clusters it is easier to estimate the quality visually; a large number of clusters can give a finer separation.
   5) Last we need to evaluate the quality of the clustering. For unsupervised methods this is sometimes hard, but in this case we can do the following: First we look if words with different senses went nicely to different clusters using the correct senses from the training set. This does not prove much, because if we chose as many clusters as we have words in the training set, we would get automatically the best result. Instead, we use the training samples to label each cluster. This means that the cluster that has more items of some sense A (*relative to the size of the training set for both senses*) alleges that all samples that go nearby have surely the sense A. We try the test set against these senses and see how many are correct.

   Using the method described above we got the results in Table 1. Here we used a map of size 9×5. If no correct answers are available, it is easier to evaluate the result when we have small number of groups. For example, for words "sade" and "komissio", the results for a 2×3 map were 59% and 98%. In Figure 2 we have the grouping of words "sade" and "komissio" for the 9×5 map.

Table 1: *Results*, 9×5 *map*

w_1	w_2	training		test	
		w_1 correct %	w_2 correct %	w_1 correct %	w_2 correct %
Lappi	Pariisi	63	55	61	53
sade	komissio	66	93	66	92
Venäjä	tammikuu	80	60	78	60
Halonen	TPS	62	74	63	70
leijona	ydinvoima	70	55	75	48

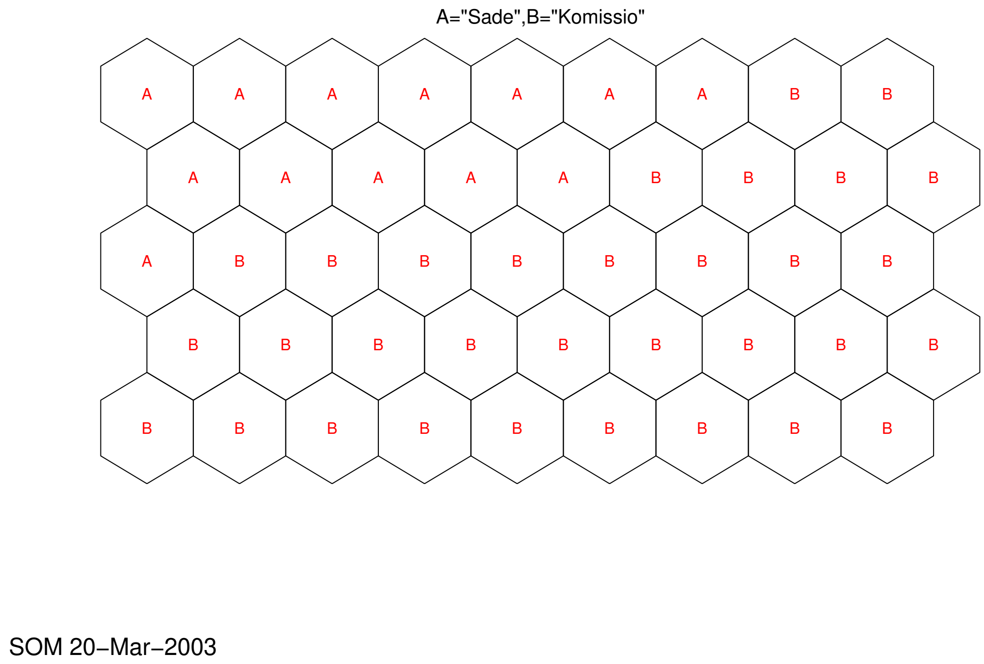


Figure 2: 9×5 *map*