Unsupervised Semantic Tagging for Swedish

Peter Floderus
Faculty of Computer Science, University of Lund, Sweden
mat05psa@cs.lu.se

Abstract

Loosely implementing a unsupervised method for semantic role labelling in Swedish first implemented by Swier-Stevenson in 2006. The task is to for a verb decide appropriate tags for nouns, the method used is an unsupervised method, which means no supplemental statistics or corpora may be used.

1 Introduction

In computer linguistics, a common problem is, given a sentence, to decide semantic roles for the different word in the sentence. This can be applied in many different areas, ranging from grammar checking in word-processors to computer parsing of newspapers, it can also be very useful in automated translation between languages. A simple method of doing this is creating a set of rules and apply them to the text. This may produce very strange results, since using too few rules may result in unspecified behaviour, and too many rules can result in contradictions. Another approach
is using an hand-annotated text to extract a base for statistics, then statistically tag other texts with the base extracted. A problem with this method is that no two texts are the same, in the most extreme cases, we use statistics derived from law-text to tag a medical report, this can be countered by using large wide-spanning corpora, but that generalizes too much. A more refined approach is to iteratively extract statistics from the text itself and then use them to tag the text. This approach is called unsupervised methods, implying that the machine works on the text without any prior input.

Swier-Stevenson applied a unsupervised method for semantic role labelling in English, since hand annotated corpora can be scarce in foreign languages (specifically Swedish), it can be interesting to implement a unsupervised method for Swedish, using valency frames from the word bank Lexin . An example for a valency frame for the word nobbar (rejecting) is A & B/x , A means a person & meaning the verb, B/x means another person or an object. The purpose of this can primarily be to get a good comparison base for other labelling methods in the language. For example, there exist today no annotated corpora for semantic roles in Swedish.

2 Related

The main related work is Unsupervised Semantic Role Labelling by Robert S. Swier and Suzanne Stevenson, which is the whole base for this work.

Other works of interest is the Talbanken corpus, a Swedish annotated corpus from the University of Lund.

The dictionary Lexin , a Swedish on-line dictionary, which defines the semantic frames the tags are extracted from.

3 Main

The task is to implement a unsupervised semantic dependency tagging algorithm on a Swedish corpus. The algorithm used is a slightly modified version of the one used by S-S since the corpus used contains additional information, like semantic dependencies. The outline of the algorithm consists of an initial rule guided tagging. Then iterating over the set of untagged words, using statistics derived from the initial tagging, tagging words with high enough probabilities.

3.1 The Corpus

The corpus used is the Talbanken corpus from Lunds University converted to standard txt format instead of xml. The corpus is consists of a 6316 sentences of both written and spoken Swedish, an example of a sentence from the corpus looks like this.

1	Individuell	jj	2	ATT	
2	beskattning	nn	0	ROOT	
3	av	pp	2	ATT	
4	arbetsinkomster		nn	3	PR

(Individual taxation of work related income)
Where the first number is just a index, the second is the word in the form it's written/spoken in the sentence. After follows the POS tag, the dependency for the word, the dependency tag and if tagged, a tag. for example the word,

3	av	pp	2	ATT

is the third in the sentence, the word av (of) is a preposition hence the POS tag pp , it's dependant of the second word beskattning and has the tag ATT meaning attribute .
A tagged word looks like this,

2	befolkningen	nn	4	SUB	x

with the tag x in the end.

3.2 Lexin Parsing

The Lexin dictionary contains, for each word, in base form, a set of words in different forms, an entry of the Part of Speech and if a verb the valency frames in wich it is used. Each form of the word is given it's own valency frame.
The frame syntax defines two kinds of elements, persons and objects, both nouns.
Exceptions, like a requirement for a place in a frame, are written in capitalized Swedish.
Optional parts of a frame is contained within parenthesis. Required, but interchangeable objects are spearated with a / sign.
If a certain word is required its written out in un-capitalized Swedish.
For example, the frame for the word ankommer (arrives) the frame is designed as this.
A/x & + TID (till + PLATS)
A/x requires either a noun, person or object, for example jag (I). TID requires a time indication, for instance 11.00 . The (till + PLATS) part is, since the bracketing, optional, but it defines that if the PLATS (place) part should be included there must also be a till (to), to complete the place indication.
This could for example be done with till Lund (to Lund), creating the complete sentence jag ankommer 11.00 till Lund (I arrives at Lund by eleven).
The frame parsing is implemented in Java using the SAX engine. A data type frame is defined as a list of elements from the actual frame, all parenthisation is discarded.

3.3 Initial frame matching

An initial discard phase, discarding all verbs without a mapping in Lexin, throws away all un-notable words and then add the remaining to a set called unnotated . The now remaining sentences have, for each verb, a large set of candidate frames. Each frame is controlled if it's plausible to tag with, looking at the direct dependencies, if a sentence have less dependence than the frame has slots, the frame is discarded as a candidate.
For all elements in the un-notated set with exactly one frame candidate, match as closely as possible to the dependencies in the corpus. For example, in the sentence (from the corpus)

1	Psykologerna	nn	2	SUB	A
2	utarbetar	vb_fin	0	ROOT	
3	allt	ab	6	DET	
4	fler	jj	3	ID	
5	förfinade	jj	6	ATT	
6	test	nn	2	OBJ	x
7	för	pp	2	ADV	
8	att	ie	7	PR	
9	få	vb	8	IM	
10	rätt	nn	11	DET	
11	man	pn	9	OBJ	
12	placerad	jj	9	OBJ	
13	på	pp	9	ADV	
14	rätt	jj	15	DET	
15	plats	nn	13	PR	
16	i	pp	15	ATT	
17	produktionen	nn	16	PR	
18	.	mad	2	IP	

the verb utarbetar has only one frame, (A & x) and the two matching nouns has been identified and tagged. The sentence is then moved from the set, unnotated to the set newly notated .

3.4 Weka statistics

For each tagged word token, the dependant verb, the Part of speech, the dependence tag and the tag itself is inserted into a weighted decision tree implemented in Weka. When given a query asking for different tag probabilities for a certain word, given all supplemented info, Weka returns a set of probabilities for the different tags. The classifier used for calculating the probabilities is a *naive bayes* classifier, this always return probabilities > 0 for all elements, even those unseen.

3.5 Statistical tagging

The method for tagging the sentences the initial frame matching didn't handle, we use the statistics Weka supplies us with. Given a threshold value for probabilities *(P(Tag|Verb,POS,Dep))*, we choose the tag with the highest probability and a probability that is higher than the threshold value. If this fails, we use a backoff model , remove the verb from the probability model and try again *(P(Tag|POS,Dep)* thus ignoring which verb seen in the sentence to raise the probability.
If still no such tag exist the sentence remain in the unnotated set. If after a full iteration cycle no sentences have been tagged, the threshold value is decreased. This iteration continues until either threshold value < 0.01 or all sentences has been tagged, whichever comes first. That the threshold value is less than 0.01 means that the probability for a given tag
(P(Tag|Verb,POS,Dep)) is less than one percent.

3.6 Implementation

The algorithm have been implemented using the Java API.

4 Results

Of the 6316 sentences, 3044 where immediately discarded due to un-existing listing in Lexin. In the initial tagging 640 were tagged, and the probability iteration tagged 119 more, resulting in 759 tagged sentences altogether.
When i refer to the document i refer to the output consisting of only the sentences from the corpus where tags have been applied.

4.1 Simple Frames

One verb of interest is the word kommer that, in the initial tagging is assigned the valency frame A/x & x, for instance corresponding to Jag kommer hem (I arrive at home), with perfect sense.
Due to lack of rules in the parser, the A is always chosen before the x, this results in a large group of sentences with a incorrect A as the subject, for example.

1	Rapporten	nn	8	SUB	A (should be an x)
2	om	pp	1	ATT	
3	gifthalterna	nn	2	PR	
4	i	pp	3	ATT	
5	de	dt	7	DET	
6	här	ab	5	ID	
7	fisksorterna	nn	4	PR	
8	kommer	vb_fin	0	ROOT	

9	från	pp	8	ADV	
10	en	dt	12	DET	
11	omfattande	pc	12	ATT	
12	undersökning	nn	9	PR	x
13	som	hp	14	SUB	
14	gjorts	vb	12	ATT	
15	av	pp	14	ADV	
16	en	dt	17	DET	
17	forskargrupp	nn	15	PR	
18	på	pp	17	ATT	
19	Riksmuseet	nn	18	PR	
20	.	mad	8	IP	

Similarly, the statistical tagger have returned a high probability for this example.

1	Ett	dt	3	DET	
2	staligt	jj	3	ATT	
3	lekråd	nn	4	SUB	A
4	kommer	vb_fin	0	ROOT	
5	att	ie	4	VC	
6	inrättas vb	5	IM		
7	under	pp	6	ADV	
8	1971	rg	7	PR	
9	.	mad	6	IP	

Where the verb kommer always have a A for the subject, the nn tag (for noun) just confirms this.

The rule tagger doesn't have a obvious target for the object (the x in the A & x frame) which rules out a tagging in the initial phase of the algorithm. These sort of results for the word kommer is very common among the tagged sentences. Either a A tag where a x would be appropriate or a random A among subjects (SUB) words, which is a result from the dropping of the verb part in the statistical analyser.

4.2 Complex Frames

The more complex tags are scarce throughout the document, partly due to the rarity for the appearance of complex tags in Lexin, but equally due to the dropping of complex frames in many cases in the initial tagging. Places where the more complex frame have been used still exist with rather bad results, for example.

1	Det	pn	2	SUB	x
2	är	vb_fin	0	ROOT	
3	lätt	jj	2	PRD	
4	att	ie	2	SUB	
5	förstå	vb	4	IM	
6	,	mid	5	IP	
7	att	sn	5	OBJ	
8	barn	nn	16	SUB	
9	som	hp	10	SUB	
10	har	vb_fin	8	ATT	
11	det	pn	10	OBJ	
12	bra	ab	10	OBJ	
13	med	pp	10	ADV	
14	sina	ps	15	DET	
15	närmaste	jj	13	PR	
16	blir	vb_fin	7	UK	
17	mer	ab	18	ADV	
18	förtroendefulla	jj	16	PRD	
19	och	kn	18	CC	
20	vänliga jj	19	CC		
21	mot	pp	16	ADV	
22	andra	jj	21	PR	
23	än	kn	22	ATT	
24	de	pn	23	UK	
25	som	hp	27	SUB	
26	ständigt ab	27	ADV		
27	upplever	vb_fin	24	ATT	
28	otrygghet	nn	27	OBJ	att +SATS
29	och	kn	28	CC	
30	bristande	pc	31	ATT	
31	förståelse	nn	29	CC	
32	.	mad	2	IP	

Where two frames have been applied, one frame from one of the words är , förstå , har and blir , specifically from blir with the frame (x &). Which resulted in the initial x .

The last verb upplever have three possible frames,

(A & x/att + S), (A & B/x som + PRED) and (A & att+SATS) the two first have been discarded due to syntactic parsing problems, which is discussed in the conclusion. The frame (A & att +SATS) have been used, since one candidate already have been tagged (with an x) the first A tag is ignored, and the last att+SATS part interprets as one tag resulting in the tagging above.

In contrast, the sentence.

1	Med	pp	4	ADV	
2	den	dt	3	DET	
3	ökningstakten	nn	1	PR	A
4	skulle	vb_fin	0	ROOT	
5	vi	pn	4	SUB	+
6	bli	vb	4	VC	
7	i	pp	10	ADV	
8	runt	pp	7	ID	
9	tal	nn	8	ID	INF
10	1950000	rg	11	DET	
11	människor	nn	6	PRD	
12	år	nn	13	DET	
13	1985	rg	6	ADV	
14	.	mad	6	IP	

Have been tagged using the frame (A/x & + INF). Which makes a tag out of the + sign.

The overall result is that most of the sentences tagged using the initial rule tagging have some extremely bad results (as exampled above). And many rather close to correct, which means a shift between A and x or correctly tagged sentences, as exampled with the kommer example.

The statistical part of the tagger generates more partly tagged sentences, that have mostly correct, or close to correct (A/x shift) tags, but incomplete tagging.

5 Conclusions

The two main problems in retrospective is primarily the insufficient syntax parsing and lack of rules to implement the frames. Secondarily, the insignificance of the statistical model.

5.1 Lexin syntax

The Lexin parser doesn't fully utilize the detail that can be derived from the syntax in Lexin, this prevents the more sophisticated frames from ever being taken into account. A more sophisticated syntax parser would be the highest priority for a future development of this. A problem with the Lexin syntax is the inconsistency of + signs and the mixing of semantic labels and actual words, for instance the att+SATS compared to the (A/x & + INF) frame, with different syntax.

5.2 Frame rules

The original algorithm by S-S uses a scoring system over the POS-chunks to determine the most probable frame(s), a similar system over the dependencies greatly favours small, simple frames since more complex frame structures often have dependencies embedded further down in the dependency tree, thus is more difficult to define. One possibility to solve this would be to device a rule system to determine the role class for every word, or at least every candidate word for a slot in a frame. This could in the most common example be a system to resolve the A/x problem, which in the current system always resolve to an A. This would require a separate program to decide whether a noun is a person or an object, with more sophisticated tags it would require a large portion of work (perhaps a separate project?).

5.3 Threshold values

The statistical tagger uses threshold values to determine which tags are eligible at a given point. Without the backoff model this would be completely useless, since a tag only can reinforce its own probabilities, the backoff model partly eliminates this, but in practice the backoff model didn't create any detectable difference in the model (5 extra tags). This can either be caused by bad implementation in the algorithm, or insignificance from the backoff model.

If the backoff model is dropped, since now a tag only can reinforce it's own probabilities, the significance of the system using threshold values is completely eliminated. No difference between a system using threshold values and one not are detected except for the order in which the taggings occurs.

This result again verifies the significance of the rules tagging model, without the backoff model, the statistical tagger just verifies the initial results. With the backoff model the tagger manage to create 5 different tags from without the model, one interesting note is that the number of tags are equal in the both cases, so the difference between the two models is caused by the threshold value.

The use of threshold values is more common among randomized algorithms, a future development method would be to use different threshold decreasing methods and compare the final results for them to determine, for each different text, the optimal model.
This would of course require a more sophisticated testing system to decide results in an automated way.

6 Acknowledgements

I would like to thank Richard Johansson for his dedication and help, and Pierre Nugues for his valuable comments on this work.

7 References

R. Swier & S. Stevensson, Unsupervised Semantic Role Labelling, University of Toronto. 2004.

Lexin, http://lexin.nada.kth.se/

Talbanken, http://w3.msi.vxu.se/~nivre/research/talbanken.html