EDAN20
Final Examination

Pierre Nugues

October 18, 2011

The examination is worth 167 points. The distribution of points is indicated with the questions. You need 70% to have a mark of 4 and 85% to have a 5.

1 Closed Book Part: Questions

In this part, no document is allowed. It is worth 73 points.

Chapter 1. Cite three applications where you think language technology plays a significant role. 3 points

Chapter 1. Describe the purpose of the operations shown in Fig. 1 respectively from step 1. to 2. and from step 2. to 3 mean. 4 points

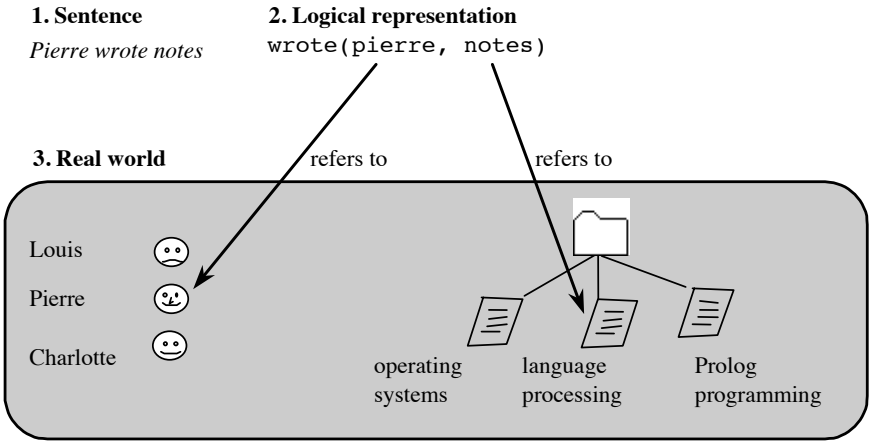

Figure 1: Operations in language technology.

Chapter 1. Name typical components of processing pipeline that could be used in the conversion from 1. to 2. in Fig. 1. You can think of the components used the the Persona system by Microsoft (Peedy) or the steps (columns) used the CoNLL evaluations. 4 points

Chapter 2. Identify what the regular expressions in the list below will match in the text (identify all the matches unless specified): 10 points

> Den här veckan kommer troligen hösten till hela Skåne. Ett blandat väder med den hel del sol, men också regn och vindar upp till 15 meter i sekunden i byarna är att vänta. Vi får fler och fler rapporter från stationerna i Skåne att hösten är här, säger Linnea Rehn, meteorolog på SMHI. Troligtvis har hösten fått fäste i hela Skåne när veckan är slut. Höst är det när dygnsmedeltemperaturen rör sig mellan noll och tio grader.
>
> *Sydsvenskan.se*, Retrieved October 16, 2011.

List of regular expressions. The symbol ␣ means a white space (visible white space):

1. `el*a`
2. `el?a`
3. `el+a`
4. `el{2,4}a`
5. `a.+a`
6. `a.+?a`
   Give only the first match.
7. `e\.␣`
   Note the trailing white space.
8. `(.)\1e`
9. `[a-zA-ZåäöÅÄÖ]+␣`
   Give the description of the results, not the complete list of strings.
10. `^.+?␣`

Chapter 2. Write a regular expression that will match the word *Skåne* with up to 10 characters to the left and up to 10 characters to the right. 3 points

Chapter 2. What does this Unix command do?
`tr -cs '0-9' '\n' <file` 1 point

Chapter 2. Using the three edit operations: copy, delete, and insert(letter), write the operation sequence that will enable you to transform (edit) each typo string shown in Table 1 into the correction string. Abridge the operations into c, d, and i(letter), For instance, String X to String Y: c c c d... 6 points

Chapter 3. Describe briefly what Unicode is. 2 points

Chapter 3. Describe briefly what UTF-8 is. 2 points

Chapter 3. Describe in general terms the structure of an ARFF file, the header and the data section, used by the Weka toolkit. 2 points

Chapter 3. Give the definition of the entropy of a set consisting of p positive examples and n negative examples. 2 points

Table 1: Typographical errors (typos) and corrections.

#	**Typo**	**Correction**	**Edit sequence**
1.	acress	actress	
2.	acress	cress	
3.	acress	caress	
4.	acress	access	
5.	acress	across	
6.	acress	acres	

Chapter 4. Using a unigram approximation, give the probabilistic model of the sentence *Den här veckan kommer hösten.* You will ignore possible start and end of sentence symbols. 2 points

Chapter 4. Using a bigram approximation, give the probabilistic model of the sentence *Den här veckan kommer hösten.* You should have exactly the same number of terms as in the previous question. You will ignore possible start and end of sentence symbols. 2 points

Chapter 4. Using the counts in Table 2, you will compute the probability of the sentence *Den här veckan kommer hösten* using a unigram and a bigram approximation. You will use fractions to represent the terms in the product and you will not try to reduce them, *i.e.* you will write $\frac{1}{3}$ and not 0.33. 6 points

You may need the total number of words in the corpus. You will use the count of *gång*, 44 500 000, with the (very rough) estimation that it represents 1‰ (1 per thousand) of the corpus.

Table 2: Word and bigram counts retrieved from Google on October 16, 2011 with the filter `site:se`.

Words	**Unigram counts**	**Bigrams**	**Bigram counts**
den	252 000 000		
här	161 000 000	den här	62 300 000
veckan	38 900 000	här veckan	2 490 000
kommer	163 000 000	veckan kommer	338 000
hösten	24 900 000	kommer hösten	51 300

Chapter 6. Using the parts of speech: determiner, adjective, noun, pronoun, modal, and verb, and the sentence: *That round table might collapse*, annotate each word with 4 points

1. its correct part of speech;
2. an incorrect but possible part of speech given by a lookup in a dictionary.

Chapter 6. Tell how to disambiguate the part of speech of the word *that* in the previous question. You can imagine a method using rules you would write or derived from an annotated corpus. 2 points

Chapter 6. What is a confusion matrix? 1 point

Chapter 8. Identify all the noun phrases in the sentence in Table 3. 2 points

Table 3: An excerpt from the CoNLL 2000 corpus.

Words	**Parts of speech**
He	PRP
reckons	VBZ
the	DT
current	JJ
account	NN
deficit	NN
will	MD
narrow	VB
to	TO
only	RB
#	#
1.8	CD
billion	CD
in	IN
September	NNP
.	.

Chapter 8. Using the parts of speech in the second column, write phrase–structure rules or regular expressions that will identify the noun phrases in the sentence shown in Table 3. 2 points

Chapter 10. Represent graphically the dependency graph of the sentence *Den här veckan kommer hösten* using data in Table 4. 2 points

Table 4: Dependency analysis of *Den här veckan kommer hösten* in the CoNLL format.

Index	Word	Head
1	den	3
2	här	3
3	veckan	4
4	kommer	0
5	hösten	4

Chapter 11. Define the four actions – shift, reduce, left-arc, and right-arc – used in Nivre's parser to parse a dependency graph and create arcs. 4 points

Chapter 11. Using gold-standard parsing and the Nivre parser, parse the sentence *Den här veckan kommer hösten* and give the sequence of actions to complete the parse. You will represent graphically the stack and the queue with the words they contain. 5 points

Chapter 15. Describe the most common structure used in simple dialogue systems. 2 points

2 Problem

In this part, documents are allowed. It is worth 94 points.

The objective of this part of the examination is to investigate and program a coreference solver. We will use Soon et al. (2001)'s algorithm described in the textbook in Sect. 14.7.4 and the corpus format provided by the CoNLL 2011 evaluation (Pradhan et al., 2011).

2.1 The Format

The CoNLL 2011 corpus consists of a set of documents in English, where the coreference chains are marked in each document. The CoNLL 2011 format contains syntactic and semantic annotations. It uses columns to describe the index, words, lemmas, parts of speech, parse trees, semantic types and structure. The coreference chains are given in the last column. We will use a simplified version of it in this examination. The (simplified) annotation of the sentence

> " Vandenberg and Rayburn are heroes of mine , " Mr. Boren says , referring as well to Sam Rayburn , the Democratic House speaker who cooperated with President Eisenhower . " They allowed this country to be credible .

is shown in Table 5. Sentences are separated by a blank line.

The goal of coreference solving is to predict the last column from the other columns given as input. The CoNLL 2011 corpus is divided into two sets: the training set and the test set. The training set contains all the columns so that the participants can develop and train a coreference solver. In the evaluation step, a test set is provided to the participants that contains all the columns, except the last one. The coreference solvers have then to predict this last column. Each participant applies her/his system to the test set and returns it to the organizers that will evaluate its accuracy using a predefined metric.

2.1.1 Understanding the Format: Coreference Chains

The CoNLL format lists coreference chains containing at least two expressions (or mentions) referring to a same entity. List all the entities with their mentions (referring expressions) occurring in the two sentences in Table 5. Entity mentions possibly embed mentions.

You will use the notation Entity(X) = {Mention1, Mention2, etc.} 3 points

2.1.2 Understanding the Format: Singletons

Entities referred only once are called singletons and are not marked in CoNLL 2011. This means that entity 6 in Table 5 is not a singleton and that the document contains at least one other mention of it in a part not shown in this table.

Identify one singleton that is obviously an entity in the excerpt shown in Table 5. 2 points

Table 5: Simplified annotation of two sentences in the CoNLL 2011 corpus. After Pradhan et al. (2011).

Document	Inx	Word	POS	Parse bit	Type	Chain
wsj_0771	0	“	“	(TOP(S(S*	*	-
wsj_0771	1	Vandenberg	NNP	(NP*	(PERSON)	(8|(0)
wsj_0771	2	and	CC	*	*	-
wsj_0771	3	Rayburn	NNP	*)	(PERSON)	(23)|8)
wsj_0771	4	are	VBP	(VP*	*	-
wsj_0771	5	heroes	NNS	(NP(NP*)	*	-
wsj_0771	6	of	IN	(PP*	*	-
wsj_0771	7	mine	NN	(NP*))))	*	(15)
wsj_0771	8	,	,	*	*	-
wsj_0771	9	”	”	*)	*	-
wsj_0771	10	Mr.	NNP	(NP*	*	(15
wsj_0771	11	Boren	NNP	*)	(PERSON)	15)
wsj_0771	12	says	VBZ	(VP*	*	-
wsj_0771	13	,	,	*	*	-
wsj_0771	14	referring	VBG	(S(VP*	*	-
wsj_0771	15	as	RB	(ADVP*	*	-
wsj_0771	16	well	RB	*)	*	-
wsj_0771	17	to	IN	(PP*	*	-
wsj_0771	18	Sam	NNP	(NP(NP*	(PERSON*	(23
wsj_0771	19	Rayburn	NNP	*)	*)	-
wsj_0771	20	,	,	*	*	-
wsj_0771	21	the	DT	(NP(NP*	*	-
wsj_0771	22	Democratic	JJ	*	(NORP)	-
wsj_0771	23	House	NNP	*	(ORG)	-
wsj_0771	24	speaker	NN	*)	*	-
wsj_0771	25	who	WP	(SBAR(WHNP*)	*	-
wsj_0771	26	cooperated	VBD	(S(VP*	*	-
wsj_0771	27	with	IN	(PP*	*	-
wsj_0771	28	President	NNP	(NP*	*	-
wsj_0771	29	Eisenhower	NNP	*))))))))))	(PERSON)	23)
wsj_0771	30	.	.	*))	*	-
wsj_0771	0	“	“	(TOP(S*	*	-
wsj_0771	1	They	PRP	(NP*)	*	(8)
wsj_0771	2	allowed	VBD	(VP*	*	-
wsj_0771	3	this	DT	(S(NP*	*	(6
wsj_0771	4	country	NN	*)	*	6)
wsj_0771	5	to	TO	(VP*	*	-
wsj_0771	6	be	VB	(VP*	*	(16)
wsj_0771	7	credible	JJ	(ADJP*)))))	*	-
wsj_0771	8	.	.	*))	*	-

2.1.3 Understanding the Format: Mentions

In CoNLL 2011, all noun phrases are potential mentions of entities. In addition, the annotators created some additional mentions when appropriate.

1. Try to justify why noun phrases are related to entities. 2 points
2. Extract the 10 first noun phrases in Table 5. These phrases are marked with the brackets `(NP ...)` in the 5th column, where `*` denotes a word. As for constituent parse trees, these phrases are possibly embedded into other phrases and possibly embed some phrases. 5 points
3. The CoNLL 2011 organizers used a named entity tagger to detect names (6th column) and the annotators created some entities from them[1]. Cite the two mentions corresponding to named entities that are not marked as NPs in Table 5. 2 points

CoNLL 2011 contains a few chains involving verbs as in

> Sales of passenger cars **grew** 22%. **The strong growth** followed year-to-year increases.
> After Pradhan et al. (2011).

We will ignore them in the rest of this examination.

2.2 Soon et al. (2001)'s Algorithm

Soon et al. (2001)'s uses machine–learning techniques to generate a training set of coreferring (positive) and noncoreferring (negative) pairs of noun phrases. In this exercise, you will extract the pairs and their class and two features: a Boolean reflecting the string match – the two pairs are equal – and a Boolean telling if the second in the pair is a pronoun.

You will create manually an ARFF file of positive and negative examples shown in Table 5. A pair of coreferring noun phrases will belong to a class named TRUE and a pair of noncoreferring nouns phrases will belong to the FALSE class.

Positive examples. List the adjacent pairs of coreferring noun phrases and extract manually their features. 6 points

Negative examples. List the negative pairs intervening between the two noun phrases in chain 23 and extract their features. Negative examples consist of a noun phrase intervening between the two noun phrases in chain 23 and the second term of chain 23. 8 points

2.3 Resolution Program: Generation of a Training Set

2.3.1 Loading the corpus

In this question, you will design and write a program to load a CoNLL 2011 corpus. Using the Java language is recommended. You can imagine that the corpus is limited to the two sentences in Table 5.

[1] NPORP: nationalities, organizations, religions, and political parties.

1. Propose class structures, preferably in Java, to represent a word, a sentence, and a document. 6 points

2. Write a program to load the corpus using these classes. 15 points

2.3.2 Detecting the Mentions

In this question, you will detect the mentions using the noun phrases.

1. Propose a class structure to represent the noun phrases. Make provision for a field representing a possible coreference chain. 5 points

2. Complement the program in Sect. 2.3.1 so that you extract all the mentions from all the noun phrases using the 5th column. You will ignore the fact that mentions can correspond to types in the 6th column. 15 points

3. Using the 7th column, complement the program so it examines all the noun phrases and possibly assigns them with their possible coreference chain. 5 points

2.3.3 Generating the Positive and Negative Examples

We will use Soon et al. (2001)'s algorithm to generate the positive and negative examples: *a pair of noun phrases corefers* and *a pair of noun phrases does not corefer*. You will extract two features only: string match and the second in the pair is a pronoun as in Sect. 2.2.

Positive examples. Complement your program so that it extracts the features of all the adjacent pairs in the coreference chains. 10 points

Negative examples. Complement your program so that it extracts the features of all the negative pairs intervening between two coreferring noun phrases. A negative pair is formed with one of the intervening noun phrase and the second term of the positive pair. 10 points

2.4 Resolution Program: Applying the Classifier

You can now train your classifier and apply it to the test set. However, the examination time is probably over. Outline how you would implement Soon et al. (2001)'s algorithm to identify the coreference chains in the test set.

This question will give bonus points.

References

Björkelund, A. and Nugues, P. (2011). Exploring lexicalized features for coreference resolution. In *Proceedings of the 15th Conference on Computational Natural Language Learning (CoNLL-2011): Shared Task*, pages 45–50, Portland, Oregon.

Johansson, R. and Nugues, P. (2008). Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of CoNLL-2008: The Twelfth Conference on Computational Natural Language Learning*, pages 183–187, Manchester.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. (2011). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.