Statistical Text Analysis Using English Dependency Structure

Jacobo Rouces

Kongens Lyngby 2012
IMM-MSc-2012-56

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-MSc-2012-56

Summary (English)

In this thesis we design, implement and test a system for statistical text analysis that, unlike the traditional bag-of-words model, takes into account natural language syntax. Our working hypothesis is that there is information held by the syntactic structure of a text, that is permanently lost when using the bag-of-words model, and which may be relevant when judging semantic similarity between different texts.

Therefore, we attempt to extract and use the actual syntactic structure of text. Namely, we use the English dependency structure of a large corpus, obtained with an unsupervised parser and a coreference resolution system. We translate the labelled dependency relations and the coreference information to a more semantically oriented language, which we call Entity-Property Language (EPL), and then from this we build both a term space and a document space over which respective metrics are defined. Different versions of the inter-document metric, some making use of the inter-term metric, are used in an information retrieval task, and its performance compared to the bag-of-words model. For testing we use a corpus developed from the English and Simple English Wikipedias, and a topic-based relevance measure.

We obtain a slight but statistically consistent improvement over the bag-of-words model, specially for long queries, and we suggest lines of research that may lead to further improvements, both in the inter-document metric for information retrieval and the inter-term metric for automatic thesaurus construction.

Summary (Danish)

Denne afhandling omhandler design, implementering og test af et system til statistisk tekstanalyse der, i modsætning til den traditionelle "bag-of words"model, tager hensyn til syntaks. Det opstilles en hypotese om at syntaktisk struktur i naturlig forekommende tale indeholder information som går tabt i bag-of-words modellen, og som kunne være relevant for bedømmelse af den semantisk similaritet mellem forskellige tekster.

Derfor forsøger vi at udtrække og anvende den givne syntaktiske struktur af tekst. Specielt udnytter vi afhængigheder i store engelske korpora. Disse afhængigheder findes ved brug af syntaktisk analyse og et system til løsning af problemer med sproglige henvisninger. Vi oversætter afhængighedsstrukturen og henvisningerne til et semantisk orientereret sprog, som vi kalder "Entity-Property Language"(EPL) og bygger på denne platform en repræsentation for dokumentanalyse. Forskellige metrikker i denne repræsentation anvendes til at løse søge-problemer og sammenlignes med bag-of-words modellen. I disse test bruger vi et data sæt baseret på Wikipedia.

Vi opnår statistisk signifikante og konsistente forbedringer i forhold til bag-of-words, specielt for søgninger baseret på lange søgestrenge.

Vi angiver et antal mulighed for at forbedringer både med hensyn til metrikker og med hensyn til modellen.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Informatics.

The thesis supervisor is Lars Kai Hansen, Professor at the Cognitive Systems Section, DTU Informatics.

I would like to thank Lars Kai for his useful remarks in both technical and non-technical aspects. I also have a special acknowledgement for Jørgen Fischer Nilsson, my tutor during the whole Master, for offering me to work with him in other projects, and putting up with my digressions. The main idea behind the Entity Property Language developed here, which is breaking down complex English sentences into a basic language, stems from our work with a natural language interface to Class Relationship Logic, in which relative clauses are rewritten to be expressed in an uniform logical form that constitutes a subset of First-Order Logic. DTU as a collective being should also be thanked for putting at my disposal, with the HPC cluster, the computing power of approximately 64 desktop workstations, with very little queue time.

Lyngby, 02-July-2012

Jacobo Rouces

Contents

Chapter 1

Introduction

In this thesis we design, implement and test a system for statistical text analysis that, unlike the traditional bag-of-words model, takes into account natural language syntax. Our working hypothesis is that there is information held by the syntactic structure of a text, that is permanently lost when using the bag-of-words model, and which may be relevant when judging semantic similarity between different texts. There are opposite views about this throughout the literature, some authors claiming that syntax serves a mere role for assisting language processing, and others presenting experiments where accounting for syntax improves correlation with human judgements [WHZ01]. Our assumption is that while there are many cases where the syntactic relations can be inferred from an un-ordered set of words, for instance because there is only one sensible or likely structure, there are other cases where there is an unavoidable ambiguity, for instance regarding *who* holds the agent and the pacient roles for an action, or *what* is attached to the negations.

One possible approach to overcome the lack of accountability of word order in the bag of words model is the use of n-grams. However, although this may capture some structural features, like collocations and phrases defined as particular sequences of words, it does not capture the individual semantic relations between words, as the use of intermediate phrases, such as dependent clauses, can set arbitrary word distances between related words.

Therefore, we attempt to extract and use the actual syntactic structure of text. Namely, we use the English dependency structure of a large corpus, obtained with an unsupervised parser and a coreference resolution system. We translate the labelled dependency relations and the coreference information to a more semantically oriented language, which we call Entity-Property Language (EPL), and then from this we build both a term space and a document space over which respective metrics are defined. Different versions of the inter-document metric, some making use of the inter-term metric, are used in an information retrieval task, and its performance compared to the bag-of-words model.

For parsing and extracting coreferences, we have used the Stanford Parser and the Stanford Deterministic Coreference Resolution System, respectively, which are included in the CoreNLP library. For training and testing the system, we have used a customized version of the English Wikipedia and Simple English Wikipedia, which contain together over 2500 million words of mostly correct and formal English, structured into articles, and are licensed under the Creative Commons Attribution Share-Alike and GNU Free Documentation licenses. Henceforth we will refer to then as enwiki and simplewiki respectively, and to the intersected enwiki and intersected simplewiki as the subsets of these corpora that share their article names.

Chapter 2 describes the successive stages for building the representation of text that includes syntactic and, as far as possible, semantic relations. Chapter 3 and 4 describe how to use this representation to build semantic spaces (for thesaurus construction) and document spaces (for information retrieval) respectively. Chapter 5 shows the obtained experimental results and chapter 6 discusses them and summarizes the conclusions. Appendix A includes additional plots from the results. Appendix B gathers most practical information regarding the implementation used to obtain the experimental results.

Chapter 2

The representation of text meaning

2.1 Syntax

Henceforth we will refer with *space of atoms* to the space of all possible basic representations of meaning for a document, like for instance the set of all the terms or stems in a lexicon when using the traditional bag-of-words model. Formally, this is different from the high-dimensional vector space that is built from it when the vector model is used, where one dimension is assigned to each atom, and the magnitude along the dimensions usually represents some count measure.

As noted in the introduction, we use the syntactic dependency structure of English text. The main reason for using this instead of the phrase structure, is that it produces a smaller space of atoms. The dependency structure of a text can be represented as a set of (labelled) pairs of words, which produces a space of atoms of size $n_{\text{labels}} \cdot {n_{\text{words}}}^2$. While this may be already too high for some purposes, it is much lower than what we would obtain from applying a similar approach to the phrase structure, which would be represented as relations between contained and containing constituent phrases, plus information pertaining the order of the contained phrases in every containing one. The length of these

phrases is potentially unbounded, and even setting a practical limit of L parts of speech, the space of atoms for the relations would easily become intractable with ${n_{\text{words}}}^{L}$ possible phrases. This might be solved selecting a word from each phrase or constituent as its head, and encoding the structure as the relations between heads. In practice, this yields similar results to the dependency structure itself, but with the labels not being as clearly defined as in most dependency grammars. Further details about the differences between phrase structure and dependency grammars are beyond the scope of this text, but can be consulted in [Sch98a] or [DMMM06].

The reason we have mentioned above for preferring smaller spaces of atoms is tractability. We will expand on this now, and introduce a further reason.

Regarding tractability, one might argue that even if the space of atoms is bigger under one representation than under another, the number of occurring instances in actual corpora might remain similar to other representation. Under this premise, if a proper sparse representation were used, there would be no advantage in using the smaller space. In practice, however, this is not true when the difference between the sizes of the theoretical spaces expand several orders of magnitude, as the indexing used in the sparse representation has to cover the entire theoretical space, growing logarithmically with its size. A statistical compression scheme could solve this to some degree but would imply additional complications. Besides, in our case, the premise of the amount of occurring instances being the same does not hold. Even though the number of non-occurring instances is higher in the case of phrase structure representation (for example, due to ungrammatical constituent phrases encoded in the phrase structure representation), so it is the number of occurring instances. An example of the latter is word order: phrase structure rules account for it, whereas the dependency structure of a text ignores it.

The last example also shows that the dependency structure representation unifies elements of the phrase structure representation, which carry the same semantics but different surface representation (in this case, word order [1]). This sort of *semantic unification* is another reason to prefer the dependency structure representation for any IR-related task in general.

Henceforth we will use the above term *semantic unification* to refer to the process of merging elements of the space of atoms that carry the same semantics, while we will say say that *semantic coherence* is improved when an element of the space of atoms whose instances could carry differing semantics is forked so that the semantics of each new element are coherent. In the case of the tra-

[1] This is why dependency grammars are more popular for describing languages with relatively free word order, such as the finno-ugric family.

ditional bag-of-words model, semantic unification and coherence are strongly related to synonymy and polysemy detection, respectively. These concepts can also be understood as decreasing Type II and Type I errors respectively, given a classifier that takes a pair of elements from the space of atoms and returns whether they mean the same or not [2], and taking as null hypothesis that they do not mean the same.

2.2 Entity Property Language

2.2.1 Objective

Prior to the vector representation, the labelled dependency pairs are converted to a more semantically oriented language consisting in unlabelled pairs of terms, where the second term represents a property of the entity represented by the first term. This, which we call Entity-Property Language (EPL), has the following objectives:

1. Reducing the size of the space of atoms from $n_{\text{labels}} \cdot {n_{\text{words}}}^2$ to ${n_{\text{words}}}^2$.
2. Unifying syntactic relations that have the same or similar semantics (entity-property), increasing semantic unification.
3. Removing others that have very different or no clear semantic translation, increasing semantic coherence.
4. Creating pseudo-terms to substitute terms engaged in relations that usually result in non-compositionality. Ideally, this should work towards both semantic coherence and unification.

Note that just using unlabelled syntactic dependencies would achieve the first objective, but would not pursue the rest. Note also that the creation of new terms implied by the last objective would increase the lexicon, working against the first one. In any case, our final implementation carries out the third step in a very limited fashion, delegating advanced proposals to further research.

[2] Of course, it is difficult to find in natural language terms or expressions that mean exactly the same. We assume an underlying equality relation that ignores some degree of subtlety, as we find in a thesaurus.

2.2.2 The underlying semantics

In this section we will temporarily depart from the dependency structure, defining a particular semantic relation and justifying its use as a fundamental building block for the meaning of text. We shall see in the next subsection how obtaining this semantic relation is quite straightforward from the dependency structure of a text.

We define the entity-property semantic relation as the one that is between the subject and the complement in a predicative copular sentence. The name stems from the fact that the subjects usually stands for an entity (what a noun refers to) and the complement for a property of this entity. We have made this choice because we have noticed that this semantic relation seems to be a main building block in language, and a significant amount of the meaning of English general text can be translated to simple copular sentences with only one noun in the subject, and either one adjective, noun or participle in the complement. The result is similar to existing data models such as Entity-Relationship, but we drop relationships between entities because the statistical handling (i.e. counting occurrences) of triples would require three-dimensional matrices. Another difference is that a term occurrence can work simultaneously as an entity and as a property, and thus a property is also an entity that may have other properties.

As an example, we take the two first sentences of the English Wikipedia article about the Wikipedia project in general (figure 2.1).

> *Wikipedia is a free, collaboratively edited and multilingual Internet encyclopedia supported by the non-profit Wikimedia Foundation. Its 22 million articles (over 3.9 million in English alone) have been written collaboratively by volunteers around the world.*

Figure 2.1: Wikipedia original text.

We now translate manually it to a proper English made mostly of copular sentences of the kind defined above (figure 2.2).

Wikipedia is free. Wikipedia is edited. The edition is collaborative. Wikipedia is multilingual. Wikipedia is an encyclopedia. This encyclopedia is in the Internet. Wikipedia is supported. Wikimedia is the support. Wikimedia is a foundation. Wikimedia is non-profit. Wikipedia is a container. Articles are contained. The articles are 22. 22 are million. Other articles are 3.9. 3.9 are million. The other articles are in English. The former articles are written. Writting is done. Writting is collaborative. Volunteers are the writers. These volunteers are around the world.

Figure 2.2: Wikipedia text converted to correct English copular sentences.

Sentences are cumbersome but still correct from a normative point of view, and most information is maintained for a human reader. For example, though contrived, it is correct to say that a number represents a property regarding size, the size having itself a property regarding units. Another non-trivial example is the translation of the past perfect tense to a property ("done"). "Wikipedia is a container. Articles are contained" comes from breaking the intermediate non-copular sentence "Wikimedia has articles", and a great loss of information can be appreciated here. We will come back to these issues later. In general this kind of normalization seems a good direction towards machine processing, breaking down the complex phrase structure of sentences to basic relations.

However, to achieve an homogeneous machine format, sentences have to be further simplified (figure 2.3). For this step, we remove articles and quantifiers, substituting an additional integer for the information they convey. A pair (<term>, <integer>) refers now to an unique referent, the process being a general (pronominal and non-pronominal) coreference resolution. In addition, words are substituted by roots, which unifies entities and properties with the same referent (e.g. "edited" and "edition") and should work towards semantic unification. Inflection information can sometimes be translated to a property, like for instance number declension (as property "many" or any other one if it is used coherently), or past tenses in a similar fashion to how past perfect was handled before. Sometimes, a term can express several properties that are different enough to justify the use of distinct terms. For example, the properties of being the agent and the patient respect to an action is significant and indeed a good example to indicate how information is lost in the bag of words model (to send compared with to be sent). Therefore, when the property is associated to a patient role, a prefix like "pat." can be appended to the root that denotes the property. A similar approach could be taken for the dative role, with a different prefix. This is a sort of artificial inflection that undoes the previous work of reducing the space of atoms, but our intuition is that it might be beneficial because of the significant difference in meaning mentioned above.

wikipedia(1) is free(1). wikipedia(1) is pat.edit(1). pat.edit(1) is collaborative(1). wikipedia(1) is multilingual(1). wikipedia(1) is encyclopedia(1). encyclopedia(1) is in.internet(1). wikipedia(1) is pat.support(1). wikimedia(1) is support. wikimedia(1) is foundation(1). wikimedia is nonprofit(1). wikipedia(1) is have(1). article(1) is pac.have(1). article(1) is many(1). article(1) is 22(1). 22(1) is million(1). article(2) is 3.9(1). article(2) is many(2). 3.9(1) is million(2). article(2) is in.english(1). article(1) is pat.write(1). write(1) is been(1). write(1) is collaborative(1). volunteer(1) is write(1). volunteer(1) is many(3). volunteer(1) is around.world(1).

Figure 2.3: Wikipedia text converted to machine EPL.

One disadvantage of using a pseudo-term for the patient role property, however, is that it becomes detached from the agent role, and the agent and patient entities lose any indirect connection too. How important is the information loss depends on the case: for many action verbs like "Borduria invades Syldavia", the information about Borduria being an invader, and Syldavia being invaded, may suffice for some purposes. Besides, from an IR point of view, many verbs may occur with no more than one agent and patient in a given document, so there would be no ambiguity. However, the loss is dramatic for very common and broad verbs, an example being in our Wikipedia excerpt: that Wikipedia is a haver/container (of something), and that articles are contained in something, is almost trivial information, the important link between Wikipedia and the articles being lost. This may be accepted as an intrinsic cost of using a single semantic relation, or what is the same, dropping triples in order to work with two-dimensional matrices.

We find appropriate to make a further digression. The behaviour displayed above for the verb "to have" is reminiscent of the copula itself, where the heaviest semantic link is not between the verb and its arguments, but between the arguments themselves. This may suggest dealing with "to have" as with "to be", translating our Wikipedia example to *wikipedia(1) is article(1)*. Considering the typical meronymic (part-of) semantics associated to phrases of the kind "NOUN have NOUN", and the typical subsumptive (is-a) semantics of "NOUN is NOUN", this is incorrect. However, in practice, the difference is blurred and the semantics of these verbs can be the same depending on the case and the language. As an example of this, compare "To be right" in English with the literal translation from Spanish and Danish, which would be "To have right" ("At have ret", "Tener raz'on"). In general, "having X" can be considered a property shared among entities that have X, and some languages will use a term $f(X)$ to denote

this property through a copular sentence "being $f(X)$". f can decline differently the same root ("to have hair","to be hairy") or map to a word with a different root ("to have courage","to be brave"). For our purposes, this means that we could simulate this f using again pseudo-terms for the patient [3] of "have", for example with a prefix "hav.", and similarly for any other verb where most semantics relayed between in the relation between agent and patient. In this case, instead of "*wikipedia(1) is have(1). article(1) is pac.have(1)*" we would have "*wikipedia(1) is hav.article(1)*".

Another situation where pseudo-terms can be created is when properties include prepositions, as dropping them certainly conflates different senses (being a table, being under a table, and being on a table). Further situations where pseudo-terms might be advisable are negation and phrasal verbs, which are not present in the Wikipedia example, but explained in the next subsection.

Our current example, after these modifications, ends up like in figure 2.3.

It should be noted that the fact that pseudo-terms do not exist in English is irrelevant for our computational purposes, as any statistical method that attempts to extract meaning of terms through patterns of co-occurence, such as semantic spaces, should work for these pseudo-terms as well. The same applies to properties like "been", which in our final version is used instead of the more human-readable "done" or "past", because these were not automatic. If the *incorrect* translations are coherent, they should acquire sense from their statistical properties in the same way normatively correct and human-readable terms do. The main problem of the use of pseudo-terms is the combinatorial explosion in the size of the lexicon.

In practice, we approximate roots with stemmed lemmas, instead of stemmed terms as it is customary in the traditional bag-of-words model. The lemmas are provided by the parser. The Lancaster stemming algorithm has been chosen, because it is very heavy and the output approximates better the linguistic concept of root, which is a minimal piece of lexical information. Stemming the lemma instead of the inflected words should be beneficial in this regard, as it succeeds to unify irregular inflections.

2.2.3 Translation from dependencies

The EPL translation of an English text is obtained post-processing its dependency structure. In order to obtain the latter, we use the CoreNLP library (ver-

[3] We use the notions of agent and patient instead of subject and object because they are semantic rather than syntactical, and independent of voice.

sion 2012-03-09) because the Stanford dependency parser included is labelled, state of the art, and provides post-processed output modes already oriented to semantic extraction [DMM08]. Namely, we will use the mode "Collapsed dependencies with propagation of conjunct dependencies", because it de-references dependencies with relative pronouns, substituting the latter with their referents. For example, for "...a company which is based...", "nsubjpass(based,which)" is substituted with "nsubjpass(based,company)", because "which" refers to "company". This mode also distributes conjunction. Besides, CoreNLP includes also a state-of-the-art coreference solver, which obtained the highest scores in both the open and closed tracks in CoNLL-2011 Shared Task [LPC+11] (Note that this coreference solver works with personal pronouns but not with relative pronouns, which are responsibility of the semantic processing just described).

Table in figure 2.4 shows the translation rules, and figure 2.5 shows examples of the dependencies, some extracted from [DMM08], others from the Wikipedia corpus, and others made up to reflect specific aspects.

The union of rule sets 1 to 7 would produce an EPL version with most of the characteristics introduced in the previous subsection. However, the use of pseudo-terms easily leads to a lexicon of unmanageable size, so in the implementation we have only kept the pseudo-terms for phrasal verbs. The 2b and 6b rule sets are substitutes of the 2 and 6 respectively, avoiding the use of pseudo-terms at the cost of lower semantic precision. Different versions of EPL will be used for building the semantic and document spaces, which we will explain in the next two chapters. However, rule sets 1 and 3, which we consider the core of EPL, will be shared by both. Rule set 8 translates information that we have deemed irrelevant for our kind of topic-based IR task, such as temporal and modal information given as properties of action events, and properties concerning quantification. It is part of the semantics of a text, however, and it could be relevant for other uses.

For most dependencies and rules, the connection with the example in the previous subsection is clear, with a possible exception of "xsubj" ($\dagger_1$ in fig. 2.4). This dependency links the head of an open clausal complement (verb or copula) with an external subject which, indirectly, is also its subject. Note that the four examples provided can be rewritten so that the dependent of "xsubj" is the proper subject of the verb.

As an additional remark, if rule set 5 were included, instances of "det(t,no)" such as in "No man flies" should be detected, and the negation could be moved to the verb ($\dagger_2$), producing "(man,no.fly)" as "Men don't fly" would. This would work towards semantic unification.

Other rules seemed appropiate for dependencies csubj, csubjpass, advmod, ad-

vcl, ccomp and xcomp, given the description in [DMM08]. However, they have been excluded from the implementation because tests performed on Wikipedia random sentences showed them to be too error-prone (for example the ones where the dependent is the head of a complex phrase), or the observed semantic relation was not as clear as in the description in [DMM08]. Also, those who are very infrequent (figure 2.6) have not been taken much into account, as they would have little impact on performance.

SD	EPL	Dependency description.
Rule set 1. Governor is property.		
agent(t1,t2)	(r(t2),r(t1))	Direct object in passive voice.
nsubj(t1,t2)	(r(t2),r(t1))	Subject in active voice.
xsubj(t1,t2)	(r(t2),r(t1))	Controlling subject [1].
Rule set 2. Governor is pacient-property w pseudo-terms.		
dobj(t1,t2)	(r(t2),'pa.'+r(t1))	Direct object in active voice.
nsubjpass(t1,t2)	(r(t2),'pa.'+r(t1))	Subject in passive voice.
Rule set 2b. Governor is pacient-property w/o pseudo-terms.		
dobj(t1,t2)	(r(t2),r(t1))	Direct object in active voice.
nsubjpass(t1,t2)	(r(t2),r(t1))	Subject in passive voice.
Rule set 3. Dependent is property.		
amod(t1,t2)	(r(t1),r(t2))	Adjectival modifier.
num(t1,t2)	(r(t1),r(t2))	Numeric modifier.
poss(t1,t2)	(r(t1),r(t2))	Possession modifier.
Rule set 4. Pseudo-term for phrasal verb.		
prt(t1,t2)	r(t1):=r(t1)+r(t2)	Phrasal verb.
Rule set 5. Pseudo-term for negation.		
neg(t1,t2)	t1:='no.'+t1	Negation.
Rule set 6. Prepositional phrase w pseudo-terms.		
prep<pr>(t1,t2)	(r(t1),pr+'.'+r(t2))	Prepositional modifier.
Rule set 6b. Prepositional phrase w/o pseudo-terms.		
prep<pr>(t1,t2)	(r(t1),r(t2))	Prepositional modifier.
Rule set 7. Noun modifier.		
nn(t1,t2)	(r(t1),r(t2))	Noun compound modifier.
Rule set 8. Irrelevant information.		
det(t1,t2)	(r(t1),r(t2))	Determiner.
aux(t1,t2)	(r(t1),r(t2))	Auxiliary or modal verb.

Figure 2.4: Conversion rules from Stanford Dependencies [DMM08] to EPL. "r(t)" is a function that returns the root of the term "t", with ":=" having preference in its definition. "+" stands for a string concatenation infix function, and single quotes denote string literals. Crossed out rules have been discarded because too many errors have been observed in tests.

• *Dependency description*		
Example sentence	**SD**	**EPL**
• *Direct object in passive voice*		
The thief has been captured by the police.	agent(capture,police)	(police,captur)
• *Subject in active voice*		
The police has captured the thief.	nsubj(capture,police)	(police,captur)
• *Controlling subject* †$_1$		
Tom likes to eat fish.	xsubj(eat,tom)	(tom,eat)
Tom hates to eat fish.	xsubj(eat,tom)	(tom,eat)
...Jay-Z had to work...	xsubj(work,jay-z)	(jay-z,work)
...Cruella attempts to steal...	xsubj(steal,cruella)	(cruella,steal)
• *Direct object in active voice*		
Bees pollinate flowers.	dobj(pollinate,flower)	(flower,pac.pollinat) / (flower,pollinat)
• *Subject in passive voice*		
Flowers are pollinated by bees.	nsubjpass(pollinate, flower)	(flower,pac.pollinat) / (flower,pollinat)
• *Adjectival modifier*		
...a famous bridge in Brooklyn.	amod(bridge,famous)	(bridg,fam)
• *Numeric modifier*		
...has 4 wheels.	num(wheel,4)	(wheel,4)
• *Possession modifier*		
...Sally's house.	poss(house,sally)	(hous,sally)
• *Phrasal verb*		
...until John gets over it.	prt(get,over)	get:=get.over → (john,get.over)
• *Negation*		
Men don't fly.	neg(fly,n't)	fly:=no.fly → (man,no.fly)
• *Prepositional modifier*		
...athletes running around the pitch.	prep_around(run, pitch)	(run,around.pitch) / (run,pitch)
• *Noun compound modifier*		
...his ninth studio album...	nn(album,studio)	(album,stud)
• *Determiner*		
The horse jumped...	det(horse,the)	(horse,the)
No man flies.	det(man,no)	(man,no) / (man,no.fly)†$_2$
• *Auxiliary or modal verb*		
John must deliver it. John had delivered the package.	aux(deliver,must) aux(deliver,had)	(deliver,must) (deliver,had)

Figure 2.5: Examples from dependencies used in the rule sets.

Dependency	Count	Dependency	Count
abbrev	< 0.01	neg	0.44
acomp	0.12	nn	9.69
advcl	0.89	npadvmod	0.14
advmod	4.12	nsubj	10.73
agent	0.60	nsubjpass	2.57
amod	7.92	num	2.07
appos	0.87	number	0.14
attr	0.03	parataxis	0.14
aux	2.89	partmod	0.81
auxpass	2.51	pcomp	0.12
cc	0.06	pobj	0.36
ccomp	1.02	poss	1.75
complm	0.38	possessive	0.03
conj.* (32)	5.25	predet	0.04
cop	3.35	prep.* (317)	17.03
csubj	0.07	prt	0.25
csubjpass	0.01	punct	0.01
dep	2.15	purpcl	0.02
det	14.07	quantmod	0.18
dobj	5.58	rcmod	1.35
expl	0.26	rel	0.09
infmod	0.18	tmod	0.30
iobj	0.08	xcomp	1.48
mark	0.63	xsubj	0.64
mwe	0.06		

Figure 2.6: Average count of dependencies per article in the intersected simplewiki. The average number of words per article is 144. The mode "Collapsed dependencies with propagation of conjunct dependencies" has been used, which produces a wide variety of 'conj' and 'prep' dependencies. These have been collapsed, and the number of varieties is between parentheses.

Chapter 3

Semantic space

In semantic spaces in general, terms are represented as vectors in high dimensional spaces, where each dimension corresponds to a term too. These sets of terms can be called represented terms and context terms respectively, and can be identical or different sets. Let m be the number of represented terms indexed, n the number of context terms, and S an $m \times n$ matrix. If r_i is the i-th represented term and c_j is the j-th context term, $S_{i,j}$ is given by some measure of contextual co-occurrence between r_i and t_j. For example $S_{i,j}$ can be equal to the number of documents, sentences, or n-word-sized windows in which r_i and t_j co-occur with. In the case the context is defined as a document, and the sets of represented terms and context terms are the same, S is indeed the covariance matrix of the document-term-frequency vector used in the bag-of-words model for document retrieval.

Once the semantic space is built from a corpus, it can be used in several ways. A common one is using the cosine distance to take some sort of similarity measure between terms. The actual semantics of this similarity will depend on the measure of co-occurrence, but when using a contextual definition, the similarity tends to be well aligned with meaning, usually gathering together synonyms as well as antonyms, resembling a thesaurus.

A general review of this family of methods can be found in [PL07] or [LB01]. The symmetric n-word-sized window definition is popular for automatic thesaurus

generation [SP97], but other choices can be found, with different effects. In [HHV10], $S_{i,j}$ is defined to be the number of times r_i precedes t_j in the whole corpus, which is equivalent to an asymmetric small window.

The semantic space can also be used for word sense disambiguation, by using not only the context of each word type in a corpus, but also the context of every word token (single occurrence). An example is in [Sch98b] and [SP95], which exploits second-order co-occurence in this manner.

3.1 Using EPL

The EPL translation of a large corpus can be used to build two semantic spaces, one using the entity root as element term and the property root as context term, and another one with inverted roles. Namely, $S^{\mathrm{EP}}_{i,j}$ can be defined as the number of times the entity i has property j in the corpus, and $S^{\mathrm{PE}}_{i,j}$ as the number of times the property i belongs to entity j, being $S^{\mathrm{EP}} = (S^{\mathrm{PE}})^t$. Though some roots will have more presence as entities or vice versa, for simplicity the sets of element terms and context terms can be considered identical.

The idea behind is that entities that share properties will be semantically similar, and the same for properties that are shared by the same entities. Ideally, we could use all the rule sets from 1 to 7, but in practice, due to the geometrical grow of the lexicon size caused by pseudo-terms, we will have to use a subset. We will go into this in more detail in section 5.2.

We will apply the cosine distance, together with two weighting functions that we call *inverse property frequency* (IPF) and *inverse entity frequency* (IEF). These are the result of importing the well-known concept of inverse document frequency to the semantic space: the idea is that properties/entities that are too common across all entities/properties, have less discriminating power and they should therefore be down-weighted. Clear examples where the down-weight would be beneficial are the property "have" and the entity "he" (if pronominal coreference substitution were not used). The resulting inter-term measures are

given in (equation 3.1).

$$
\begin{aligned}
s_{\mathrm{E}}(e_1, e_2) &= 1 - \frac{\sum_{p=1}^{n_{\mathrm{roots}}} S^{\mathrm{EP}}_{e_1,p} \cdot S^{\mathrm{EP}}_{e_2,p} \cdot \mathrm{ipf}(p)^2}{\sqrt{\sum_{p=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{EP}}_{e_1,p} \cdot \mathrm{ipf}(p)\right)^2 \cdot \sum_{p=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{EP}}_{e_2,p} \cdot \mathrm{ipf}(p)\right)^2}} \\
\mathrm{ipf}(p) &= \log \frac{n_{\mathrm{roots}}}{\sum_{e=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{EP}}_{e,p} > 0\right)} \\
s_{\mathrm{P}}(p_1, p_2) &= 1 - \frac{\sum_{e=1}^{n_{\mathrm{roots}}} S^{\mathrm{PE}}_{p_1,e} \cdot S^{\mathrm{PE}}_{p_2,e} \cdot \mathrm{ief}(e)^2}{\sqrt{\sum_{e=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{PE}}_{p_1,e} \cdot \mathrm{ief}(e)\right)^2 \cdot \sum_{e=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{PE}}_{p_2,e} \cdot \mathrm{ief}(e)\right)^2}} \\
\mathrm{ief}(e) &= \log \frac{n_{\mathrm{roots}}}{\sum_{p=1}^{n_{\mathrm{roots}}} \left(S^{\mathrm{PE}}_{p,e} > 0\right)}
\end{aligned}
\tag{3.1}
$$

Some terms may appear very few times in the corpus, either as entity, as property or as both, and high cosine similarities between these terms are not statistically reliable. For instance, two entities occurring only once in a corpus, and with the same property, would produce a perfect match of 1. A simple way to avoiding this is defining a threshold t_{EP} so that entities and properties that occur less than t_{EP} times in the corpus are defined to have zero similarity with other entities/properties.

Other approaches of using syntactic dependency relations to build semantic spaces are summarized and integrated into a common framework in [PL07]. Notable examples are [Lin98], [CM02b] and [CM02a], which use labelled dependencies. For the reasons explained in section 2.2, we expect some improvement from the semantic normalization carried out by EPL over the syntactic labelled dependencies, which are sparser and whose semantics often depend on the context (ie. they have lower semantic unity and coherence). Besides, [CM02b] and [CM02a] use shallow statistical parsers, and [Lin98] uses MINIPAR, which is a full-blown but relatively simple parser.

Performing spherical clustering on the EPL semantic space, clusters can be expected to gather semantically related terms. More specifically, it can be performed on the space of entities, after which clusters will be composed of entities with similar properties, and on the space of properties, after which clusters will be composed of properties often shared by the same entities. These clusters can be expected to align with sets of synonyms (similar to the concept of synsets in Wordnet), although antonyms may also be included if negation is not accounted for in the EPL semantic space, through pseudo-terms made of negated roots.

We discard applying S^{EP} to a method for word sense disambiguation such as in [Sch98b] and [SP95]. This is because, although exploiting second-order co-occurrence, and thus reducing sparsity, that method, as any other that should try to solve polysemy, has to consider relatedness at the level of word occurrences, which for the case of EPL may be unsuitably sparse (in EPL version we will use for the semantic space, the average number of properties/entities per entity/property token is way below one, tested for simplewiki). Besides, for our eventual IR task, using the EPL pairs as elements representing a document already performs a great deal of disambiguation on the individual roots.

Chapter 4

Document space

As baseline, we have used the bag-of-words model with the document terms filtered using the 119 stop-words list in table B.1 (Appendix B.1) and processed with the Lancaster stemmer, using the Okapi BM25 measure with IDF. This is formalized in equations 4.1 and 4.2, where tf stands for term frequency and df for document frequency. The BM25 measure has performed well in TREC evaluations [MRS08] and is implemented in state-of-the-art IR libraries such as Terrier [biba]. It is basically a generalization of the cosine distance, the parameters k_1,k_3 controlling how linear/binary are the elements of the document and query respectively, and b controlling the degree of length normalization of the document vector. For these parameters, we have used the typical recommended values as in [MRS08] ($k_1 = k_3 = 1.5, b = 0.75$).

$$\mathrm{BM25}(q,d) = \sum_{t \in q} \mathrm{idf}(t)\ \frac{(k_1+1)\mathrm{tf}(t,d)}{k_1((1-b)+b(L_d/L_{\mathrm{ave}}))+\mathrm{tf}(t,d)} \frac{(k_3+1)\mathrm{tf}(t,q)}{k_3+\mathrm{tf}(t,q)} \tag{4.1}$$

$$\mathrm{idf}(t) = \log \frac{N}{\mathrm{df}(t)} \tag{4.2}$$

After that, we have developed different inter-document measures that use the

EPL information, which are explained below. The rule sets used should at least contain those used for the semantic space, but can contain more, as it is discussed in the conclusion chapter.

- **bag-of-roots**. First, we have used the same bag-of-words model and inter-document function as for the baseline, but stemming the lemmas instead of the words. The stemmed lemmas is what we use as an approximation of the roots in EPL, but this measure does not use EPL proper. In practice the differences are minor, mostly concerning the conversion to the infinitive of irregular forms of verbs, and a significant difference in performance is not expected.

$$s_{\mathrm{bor}}(q,d) = \sum_{i}^{n_{\mathrm{roots}}} q_i^{\mathrm{roots}} \cdot d_i^{\mathrm{roots}} \tag{4.3}$$

- **bag-of-pairs**. The most simple way to account the EPL pairs for inter-document comparison is treating them as terms, creating a very high dimensional space, with a dimensionality equal to the number of roots squared. Both a simple cosine distance and the same BM25 with IDF measures can be used. The very high dimensionality and sparseness have practical implications. For example, the minimum size of the corpus to properly train the IDF is now higher. Also, three-dimensional matrices are not typically supported in libraries for sparse matrix representations [1] and their efficient implementation may pose additional challenges. If two-dimensional sparse matrices are used for the implementation, the indexing should use a type that admits integer above 2^32, as the number of roots squared will easily exceed 2^{32}. This could be solved removing enough uncommon terms, but we have found no justification for the removal of uncommon terms, with regard to retrieval performance, as uncommon terms are usually very specific and informative, this being the principle on which IDF weighting is based.

$$s_{\mathrm{bop}}(q,d) = \sum_{i}^{n_{\mathrm{roots}}} \sum_{j}^{n_{\mathrm{roots}}} q_{i,j}^{\mathrm{pairs}} \cdot d_{i,j}^{\mathrm{pairs}} \tag{4.4}$$

- **combination-of-bags**. The bag-of-pairs measure is very strict: documents that have the same proportion of pairs are very likely to be related, but it is also to be expected that the amount of documents that are related but do not share many pairs is higher than for the bag-of-words, because many shared words could be paired with other entities/properties (for example synonyms or specifiers). The simplest approach to reduce this

[1] Both Matlab R2012a and SciPy v0.11 lack support for three-dimensional sparse matrices.

strictness is to employ a weighted linear combination of the rank given by the bag-of-roots and the bag-of-pairs, as in equation 4.5. w is a parameter that can be optimized experimentally.

$$s_{\text{cob}}(q,d) = w \cdot s_{\text{bor}}(q,d) + (1-w) \cdot s_{\text{bop}}(q,d) \tag{4.5}$$

- **clustered-bag-of-pairs**. The combination-of-bags measure reduces the excessive strictness of the bag-of-pairs, but does so by giving weight to the similarity we try to improve, and not by relaxing the strictness of the comparison between the EP pairs, which is intrinsically related to its sparseness. A way of reducing such sparseness and strictness is using the clustering over the semantic space, defined in chapter 3. This is formalized in equation 4.6. Also, the time and space performance of the querying is improved with this method, since the dimensionality is reduced to k^2, and on average the number of non-zero elements is decreased too.

$$s_{\text{cbop}}(q,d) = \sum_{i}^{k} \sum_{j}^{k} q_{i,j}^{\text{pairs-of-clusters}} \cdot d_{i,j}^{\text{pairs-of-clusters}} \tag{4.6}$$

- **soft-bag-of-pairs**. Clustering establishes a binary categorization over a continuum of similarity, which implies an information loss. Also, besides the appropriateness of the optimal solution, a typical clustering algorithm like k-means is NP-complete and in instances of this size, only an approximate solution is feasible, often having to resort to multiple randomized runs to increase the quality of the solution. In this regard, equation 4.7 defines an inter-document measure that matches similar entities and similar properties, trying to tackle sparsity and synonymy too, but using the continuous similarity measure in the semantic space. Because of the soothing effect on sparseness, this method is expected to perform better for short queries.

$$s_{\text{sbop}}(q,d) = \sum_{i}^{n_{\text{roots}}} \sum_{j}^{n_{\text{roots}}} \sum_{k}^{n_{\text{roots}}} \sum_{l}^{n_{\text{roots}}} f(s_{\text{ee}}(i,k)) \cdot f(s_{\text{pp}}(j,l)) \cdot q_{k,l}^{\text{pairs}} \cdot d_{k,l}^{\text{pairs}} \tag{4.7}$$

Function f may be the identity function in $[0,1]$ or some shaping function. $f(x) = x^n$ with $n \geq 2$ would serve to downweight the effect of the similarity measure (a pessimistic approach), converging to bag-of-pairs when $n \to \infty$.

The problem in this case is that whereas the clustering improved the time performance of the querying, this method greatly worsens it. Namely, given a sparse representation of q^{pairs} (a query) and d^{pairs} (a document), collapsed into one dimension and ordered, s_{bop} and s_{cbop} would run in

linear time to the number of pairs in d^{pairs} (assuming it is longer than the query). Similarly for $q^{\text{pairs-of-clusters}}$ and $d^{\text{pairs-of-clusters}}$, being the number of pairs lower in general. On the contrary, for s_{sbop} all entities in q^{pairs} have to be semantically compared to the ones in d^{pairs}, and the same for the properties, giving a quadratic number of semantic comparisons. The semantic comparison can be reduced to constant time storing the distances in advance into a sparse matrix, but despite this, the cost is several times higher, as the two obtained similarity matrices and a third matrix of values, all of them of size $n_{\text{pairs-in-q}} \times n_{\text{pairs-in-d}}$, have to be multiplied element-wise and the elements of the result added up.

Bag-of-roots is the dot product in a high dimensional space, and bag-of-pairs and clustered-bag-of-pairs are reduced to that if the two dimensions (corresponding to entity and property indexes) of matrices q and d are collapsed into a singe dimension. Combination-of-bags relays on two dot products, and soft-bag-of-pairs can be seen also as dot product which follows a pre-processing that *densifies* collapsed matrices/vectors q^{pairs} and d^{pairs}. Therefore, BM25 can be applied to all these measures in the similar way it can be used instead of the dot product in the traditional bag-of-words model, and a similar relative improvement can be expected. Since BM25 can be reduced to a dot product after the element-wise processing in equations 4.8 and 4.9, the equations 4.3-4.7 need not be redefined. This can be used indeed to speed up query time.

$$q'_i = \text{idf}(i) \; \frac{(k_3 + 1)q_i}{k_3 + q_i} \tag{4.8}$$

$$d'_i = \frac{(k_1 + 1)d_i}{k_1((1 - b) + b(L_d/L_{\text{ave}})) + d_i} \tag{4.9}$$

Chapter 5

Results

5.1 Corpus preparation

Both for training the semantic space and testing the IR performance over the document space we have used the pages-articles dumps of both the English Wikipedia (enwiki) and the Simple English Wikipedia (simplewiki).

For most corpora used for information retrieval evaluation, queries consist of a list of keywords, usually taken from topic description. This is suitable for bag-of-words models, but doesn't allow to measure the potential advantages of grammatically well-formed English queries. Therefore, we have built a test corpus using simplewiki and enwiki as a source for queries and documents respectively, defining the relevant documents for a query as those that share their title with it. We consider that this ensures an appropriate gold standard of semantic relatedness, as articles with the same title talk about the same topic, though usually to different depth and vocabulary.

Apart from that, further modifications have been carried away. Enwiki documents larger than 1500 words have been split in chunks of approximately 100 words, and simplewiki documents above the same limit have been discarded. This is because of a limitation of the parser (see Appendix B.2), but has the additional consequence of allowing several related documents per query, which

smoothes certain precision and recall measures. The split has been carried out in a way that tries to maximize the contextual coherence of each chunk, aligning them with sections and paragraphs (see Appendix B.1). Nevertheless, an objection could be raised about whether the relevance relation still holds as an appropriate gold standard in this case, as the chunks can deal about subtopics that differ significantly from the main overall topic. For this reason, tests in section 5.3 have been carried out both for queries with 7 to 10 relevant documents associated and also for queries with only one document associated, the latter corresponding to an enwiki document shorter than 1500 words.

Article titles have been omitted from both simplewiki and enwiki articles, because they provide too obvious similarity between queries and documents. Section titles have been omitted too, because they tend to include certain terms that are unrelated to the content, such as *References*, which can represent a significant portion of the text in short articles, and producing bogus similarity.

The resulting configuration provides a reasonable gold standard of topic-wise document relatedness. However, IR queries built on the spot by users are seldom longer than a few words. For this reason, another set of queries have been built extracting the second sentence from each simplewiki article, which on average have the length and the associated extremely high sparsity of a typical query. We have chosen the second sentence because the first one usually includes the title of the article, which is usually repeated many times in the corresponding enwiki article, rendering the problem too easy again. On the contrary, the second sentence is usually related in a more indirect way (for example using hyponymy and/or hypernymy), and produces a more challenging query. Tests in section refsec:experiments:docspace consider both the complete-document and the second-sentence queries, as the former, despite what have been said, may also be relevant for other tasks such as article recommendation.

A total of 62,797 articles have been extracted and preprocessed for simplewiki (approx. 55 MB of ASCII text), and 3,489,310 articles divided in 3,884,494 chunks for enwiki (approx. 5 GB). The intersection between enwiki and simplewiki has given two sets of 51,312 articles each. In the case of enwiki, these have been divided in 115,292 chunks. Further details on the corpus pre-processing are described in Appendix B.

5.2 Semantic space

The EPL translation rules used for the semantic space are 1, 2b, 3, 4, from figure 2.4. All rule sets that introduce pseudo-terms are omitted for the lexicon

size problem, except number 4. 6b is not included as a substitution for 6, because we have judged inconvenient the semantic conflation it introduces (i.e. it is radically different being "in English" and "about English", or "under the sea" and "above the sea".). Rule set 7 has been omitted too because although in theory a noun modifier seems appropriate as a property (in "oil prices", prices are specified to be oil-related), in practice CoreNLP applies a rule where all nouns modify the rightmost noun in NP, which thorugh example examination we have notice that produces a lot of mismatches.

Figure 5.1 shows several examples of roots (approximated as stemmed lemmas), which have often the role of entity or property, and their entities or properties respectively, together with the associated cosine distances in the semantic space. Equations 3.1 from chapter 3 have been used, with intersected simplewiki as corpus. The examples look relatively good, considering we have omitted negation and other fine distinctions such as pacient and agent roles. However, they have been selected by hand, with frequent words, so the measure is statistically well founded, and non-polysemous, so the examples are more clear (note however, that the use of pairs in the document space should alleviate polysemy, because the property in a pair deambiguates the entity, and vice versa). The general quality is substantially below a hand-made thesaurus, specially for more infrequent or highly polysemous words. Setting a high t_{EP} threshold filters out some bad matches, but at the cost of other good ones.

We have also performed clustering over the same space, using spherical k-means. Some clusters are listed in figure 5.2. The same remark can be done about the hand-picked examples, with the additional problem that traditional clustering does not allow a polysemous word to belong to more than one cluster.

5.3 Document space

For simplicity, the lexicon used has been extracted from simplewiki, as terms that do not appear in any query do not affect the similarity score, except for the effects on vector normalization. Regarding the bag-of-words baseline, since BM25 uses document length for normalization instead of the vector 2-norm, the ranking will be identical, provided that the original lengths of the documents are recorded before indexing the terms. The term-wise IDF scores, on the contrary, have been trained on enwiki, as the corpus of documents is the only one that can be expected to be known by the system at query time.

For the evaluation of the different inter-document measures in the document retrieval task, we compare precision/recall curves, as well as the binary relevance

ENTITIES
hous : castl(0.66) , templ(0.62) , hotel(0.59) , fort(0.57)
comput : program(0.55) , tool(0.53) , method(0.53) , techn(0.52)
man : girl(0.76) , boy(0.75) , wom(0.68) , adult(0.68)
machin : tool(0.51) , it(0.48) , comput(0.47) , taty(0.46)
book : novel(0.84) , essay(0.81) , poem(0.79) , autobiograph(0.75)
sea : oc(0.43) , gulf(0.42) , lagoon(0.40) , katteg(0.35)
PROPERTIES
go : work(0.94) , return(0.91) , try(0.89) , die(0.89)
dark : gray(0.45) , brown(0.43) , pal(0.43) , blu(0.40)
lov : marry(0.84) , outshin(0.82) , panick(0.82) , homoio(0.82)
coward : risktak(0.52) , shutin(0.52) , esa(0.42) , let.down(0.38)
build : rebuild(0.62) , destroy(0.59) , design(0.58) , demol(0.54)
buy : lik(0.68) , need(0.67) , think(0.66) , keep(0.66)
think : believ(0.90) , liv(0.90) , talk(0.88) , lik(0.85)

Figure 5.1: Hand-picked examples of entities/properties, and their closest entities/properties in the semantic space, using equations 3.1. The associated similarities (1− distances) are between parentheses.

metrics *Average Precision (AveP)* (eq. 5.1) and *R-Precision (R-Prec)* (eq. 5.2) described in [Sak07] and [MSM99].

$$\text{AveP} = \frac{1}{R} \sum_{1 \leq r \leq L} \text{isrel}(r) P(r) \tag{5.1}$$

$$\text{R-Prec} = P(R) \tag{5.2}$$

R is the number of relevant documents for a topic, and L the size of the ranked output, with a typical upper bound of 10000 [Sak07]. $P(r)$ is the precision at rank r, namely $P(r) = \frac{\text{count}(r)}{r}$ where count(r) is the number of relevant documents within top r. isrel(r) is 1 if the document at rank r is relevant for the query, and 0 otherwise.

We use averaged AveP and R-Prec over all the queries, with the corresponding 95% confidence intervals. Since the results from both measures are consistent, in order to simplify the comparison of different configurations, we defer the R-Prec plots to Appendix A. In the figures of this section, the uppermost plot shows the performance of combination-of-pairs or soft-combination-of-pairs for different values of w, compared to the bag-of-words measure used as baseline. The performance for bag-of-pairs and bag-of-roots is included as the one of combination-of-pairs with $w = 0$ and $w = 1$ respectively. The lower-most plot

ENTITIES
steambo, cathedr, cemetery, bastid, monu, lighth, sawmil, hotel, nest, bunk, bleach, citadel, greenh, palac, can, gym, lodg, fol, chapel, dmu, spa, church, wal, velocirapt, tow, skyscrap, casino, carry, castl, inca, railroad, instal, pyramid, fortress, templ, vault, fort, hous, heng, pier, plaz, tramway, abbey, colosse
adult, boy, man, xen, wom, aristocr, girl, wizard
navy, milit, army, forc
england, spain, czechoslovak, jap, israel, denmark, morocco, germany, norway, berlin
PROPERTIES
crit, comp, regard, dead, dant, wait, both, want, transport, protect, bring, wish, believ, attract, trust, worry, oppos, suff, stop, know, enjoy, enco, talk, ther, feel, quest, car, buy, abl, keep, farm, kil, afraid, hop, interest, rid, say, sit, see, not, lik, stress, tre, com.up, support, visit, behav, read, hurt, remind, rememb, pract, surpr, drink, smok, drown, afford, scar, descend, tak.out, argu, find.out, forc, learn, pref, help, suppress, suspect
bright, bril, blu, col, red, gray
id, giv.away, sel, advert

Figure 5.2: Some clusters obtained running spherical k-means on entities/properties with$t_{\mathrm{EP}} = 10$, using and and average cluster size of 5, from the intersected simplewiki corpus.

shows the precision/recall curve for the different methods, combination-of-pairs being represented with the optimal w. The left-most point in the precision/recall curves correspond to a retrieval threshold of 1.

The EPL version used for the inter-document measures applies the same rule sets than for the semantic space (1, 2b, 3, 4), plus rule sets 6b and 7. This is because we have found that it gives consistently higher results, despite the semantic problems pointed out for rule sets 6b and 7. We omit results reflecting this point, to avoid the number of plots would becoming inconvenient.

Figures 5.4 and 5.6 constitute the core results, where our system obtains the best results for long and short queries respectively. Figures 5.5 and 5.7 use the same configurations respectively, except that use a set of queries that have associated several relevant documents each, and the results are averaged over 1000 queries instead of 2000. The former change allows AveP and the precision/recall curves to be smoother, even averaging over a smaller set of queries, but the underlying relevance relation is less reliable due to the problem of topic deviation in article chunks, pointed out in section 5.1.

Figure 5.8 uses the same configuration than figure 5.4, but without pronominal coreference substitution, so that the impact can be assessed.

Figures 5.9, 5.10, 5.11,serve to illustrate the difference between using bag-of-pairs and soft-bag-of-pairs with two different shaping functions. In these cases, we use short queries (second sentence) because due to the extremely high sparsity, it is where soft-bag-of-pairs is expected to be more necessary. We use $t_{\mathrm{EP}} = 2000$, a high threshold that is surpassed by only around 150 entities/properties, in order to be very strict with the statistical significance of inter-term similarity values grater than 0. Since soft-bag-of-pairs becomes equivalent to bag-of-pairs when $\forall i, j \; i \neq j \; \rightarrow \; s_e(i,j) = 0 \; \wedge \; s_p(i,j) = 0$, this is not expected to change the sign of the effect on performance from bag-of-pairs to soft-bag-of-pairs, but only the magnitude. Results obtained from clustered-bag-of-pairs, using spherical k-means as in figure 5.2, are consistently worse and they are omitted.

In the configurations where bag-of-pairs is tested, the set of documents on which the retrieval has been performed is the whole intersected enwiki. In the configurations where soft-bag-of-pairs is computed, due to the higher computational cost, it is the subset of documents in enwiki that are relevant for some evaluated query, averaging over 500 queries. The exception is figure 5.9, which uses bag-of-pairs but uses the same queries and documents than the ones with soft-bag-of-pairs, in order to allow comparison.

We assume performance-wise independence between parameters, as computing and displaying all the combinations would be impractical.

Table in figure 5.3 summarizes the configurations and the plots.

	1 rel. doc. per query	7-10 rel. docs. per query
queries are no shorter than 6 sentences	fig 5.4, A.1 fig 5.8, A.5 (1)	fig 5.5, A.2
queries are the second sentence	fig 5.6, A.3 fig 5.9 A.6 (2) fig 5.10 A.7 (2) (3) fig 5.11 A.8 (2) (3)	fig 5.7 , 5.7

Figure 5.3: Parameters used in different experiments, resulting configurations and corresponding plots.
(1): Not using pronominal coreference substitution.
(2): Using 500 queries and 500 documents.
(3): Using soft-bag-of-words.

The reasons for the differences (or the lack thereof) are discussed in the next chapter.

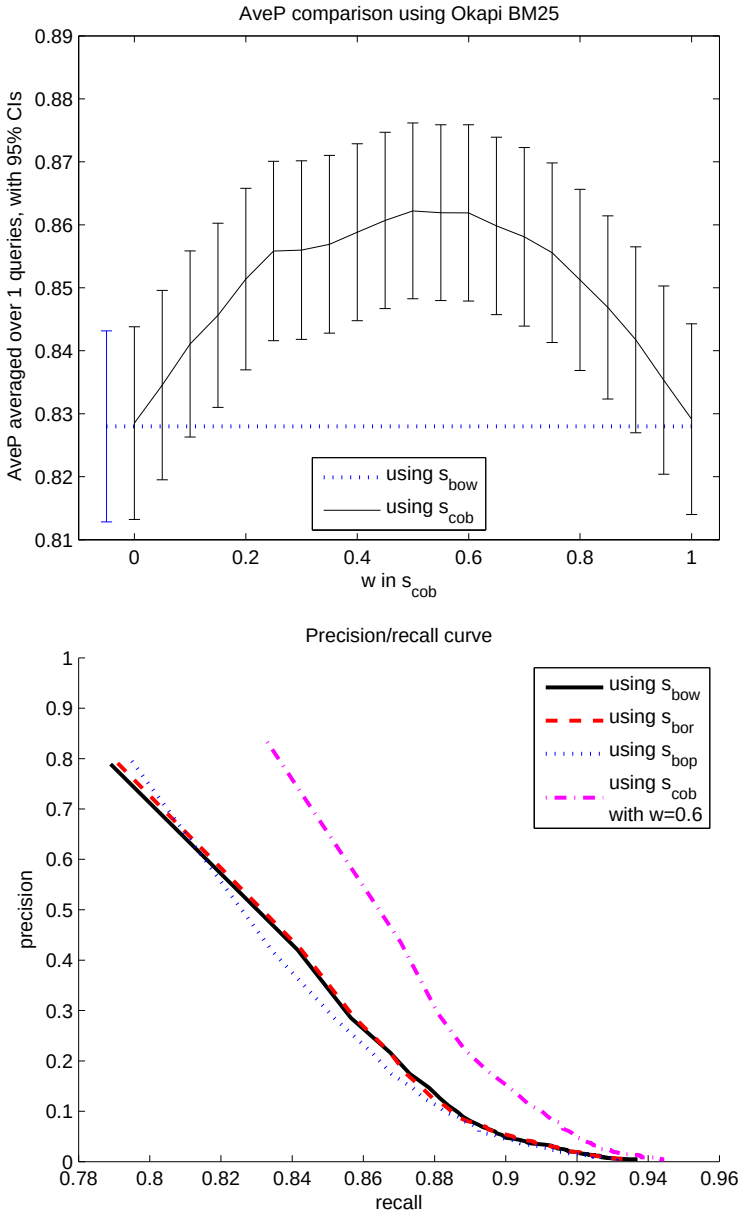

Figure 5.4: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, using pronominal coreference substitution, with queries being the complete first 2000 simplewiki articles which are no less than 6 sentences length, and have exactly 1 relevant document associated.

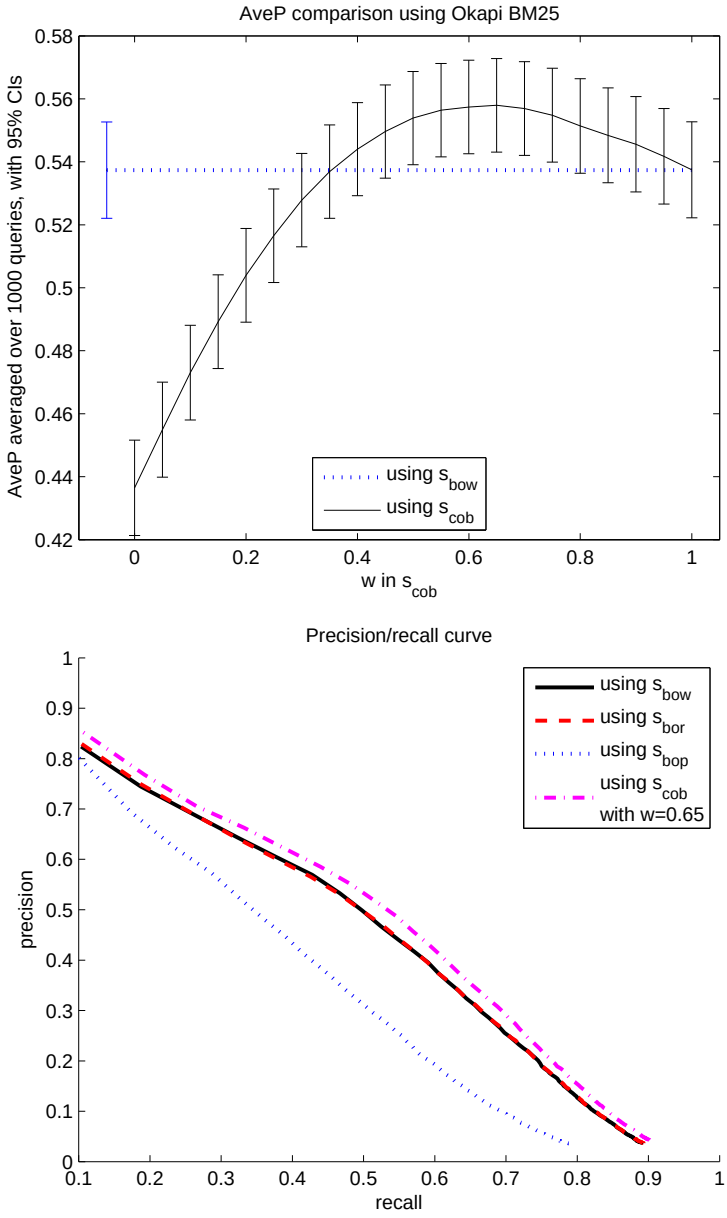

Figure 5.5: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, using pronominal coreference substitution, with queries being the complete first 1000 simplewiki articles which are no less than 6 sentences length, and have between 7 and 10 relevant documents associated.

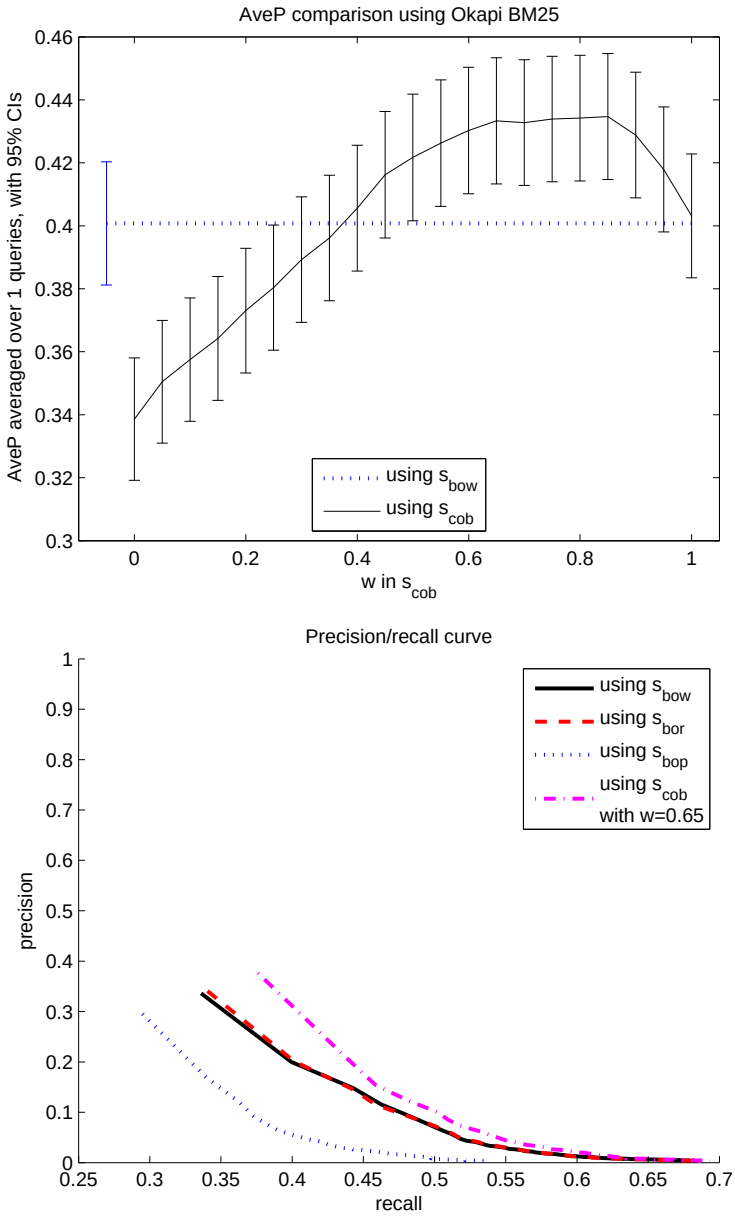

Figure 5.6: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, using pronominal coreference substitution, with queries being the second sentence from the first 2000 simplewiki articles which are originally no less than 2 sentences length, and have exactly 1 relevant document associated.

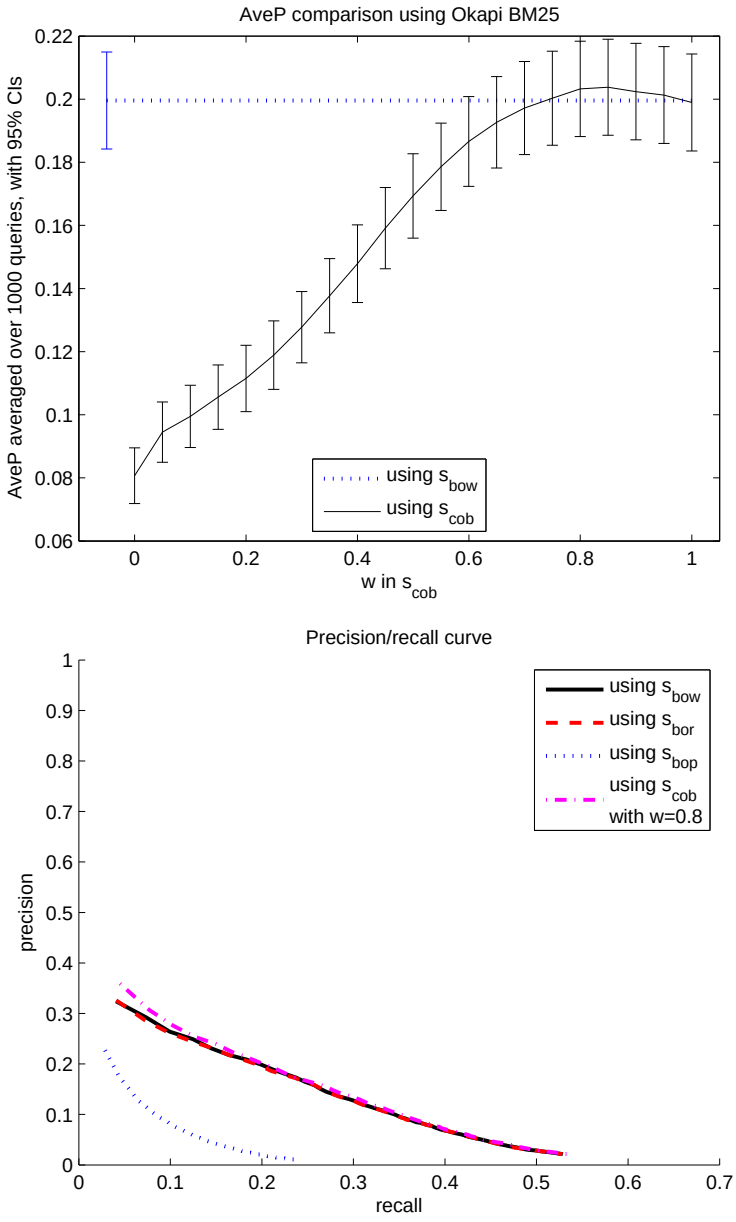

Figure 5.7: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, using pronominal coreference substitution, with queries being the second sentence from the first 1000 simplewiki articles which are originally no less than 2 sentences length, and have between 7 and 10 relevant documents associated.

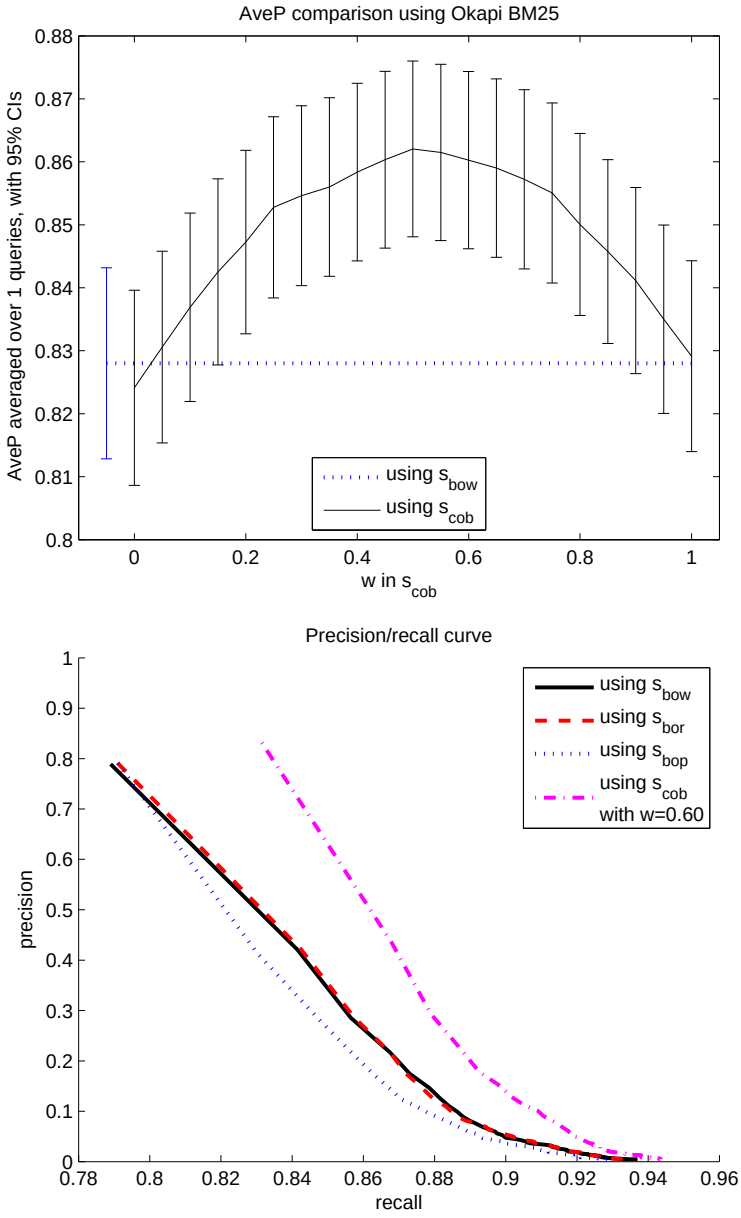

Figure 5.8: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, **without** pronominal coreference substitution, with queries being the complete first 2000 simplewiki articles which are no less than 6 sentences length, and have exactly 1 relevant document associated.

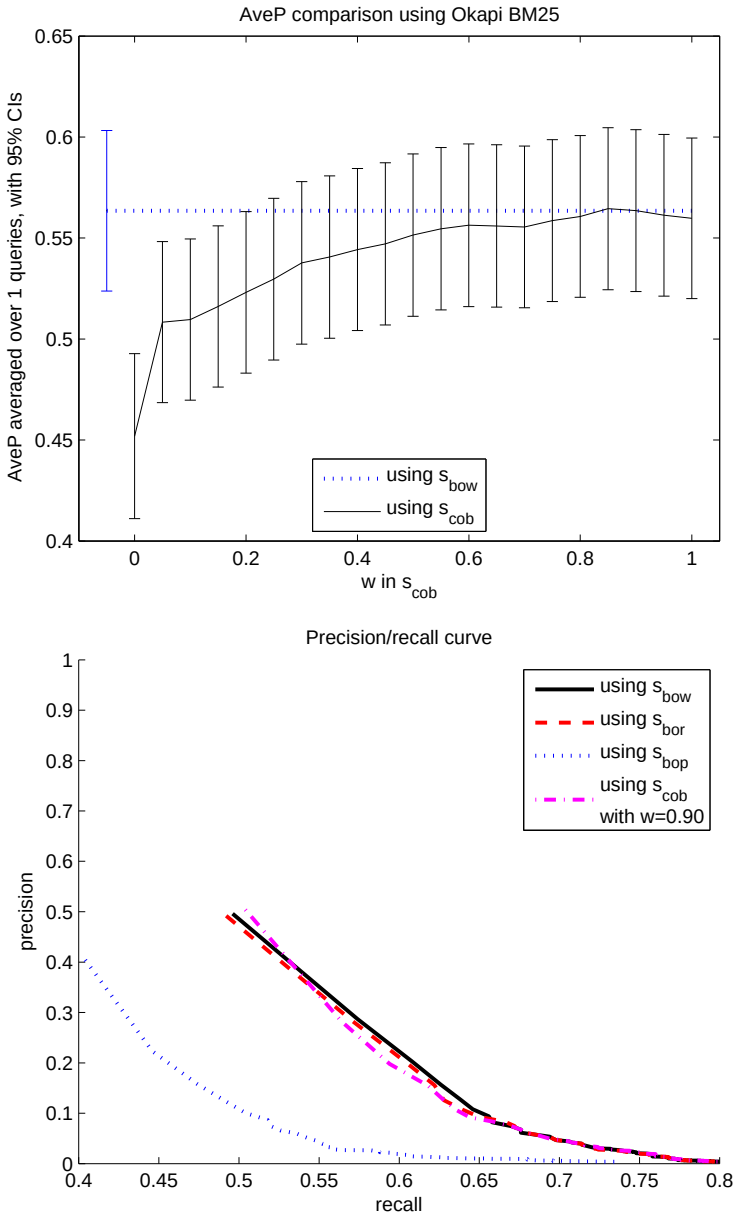

Figure 5.9: Results for bag-of-words, bag-of-roots, bag-of-pairs and combination-of-bags, using pronominal coreference substitution, with queries being the second sentence from the first 500 simplewiki articles which are originally no less than 2 sentences length, and have exactly 1 relevant document associated. The retrieval is performed only from the set of 500 documents that are relevant for some query.

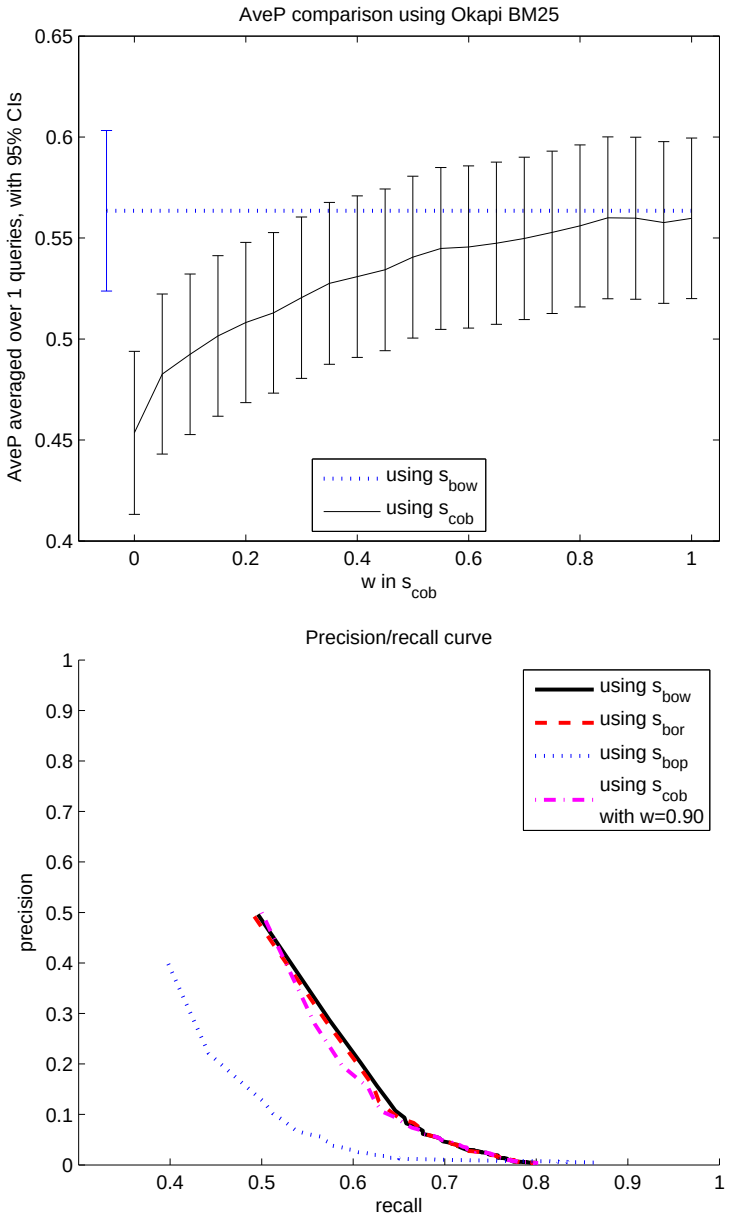

Figure 5.10: Results for bag-of-words, bag-of-roots, bag-of-pairs and soft-combination-of-bags (with shaping function $f(x) = x^2$), using pronominal coreference substitution, with queries being the second sentence from the first 500 simplewiki articles which are originally no less than 2 sentences length, and have exactly 1 relevant document associated. The retrieval is performed only from the set of 500 documents that are relevant for some query.

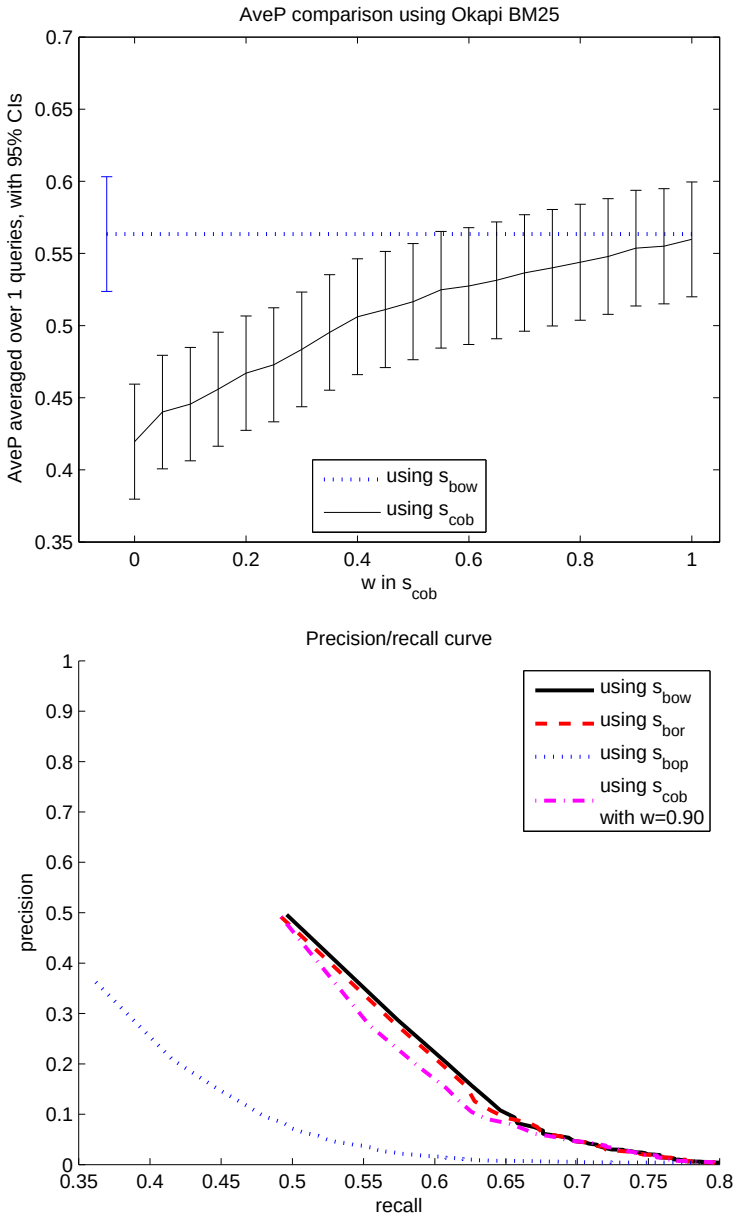

Figure 5.11: Results for bag-of-words, bag-of-roots, bag-of-pairs and soft-combination-of-bags (with shaping function $f(x) = x$), using pronominal coreference substitution, with queries being the second sentence from the first 500 simplewiki articles which are originally no less than 2 sentences length, and have exactly 1 relevant document associated. The retrieval is performed only from the set of 500 documents that are relevant for some query.

Chapter 6

Conclusion

6.1 Discussion of the results

Several inter-document measures that take into account English syntax have been developed and compared in an IR task against the traditional bag-of-words method. For this purpose, a semantic postprocessing of the syntactic information in the text has been elaborated, which we have called Entity-Property Language (EPL). From the results in section 5.3 we can make the following conclusions.

1. First, the bag-of-roots method yields almost identical performance as bag-of-words (compare fig. 5.4 and 5.8), as we initially expected.

2. The bag-of-pairs method yields the best results using long queries (no shorter than 6 sentences), in which case its performance is around the same than the bag-of-words, if it is evaluated over queries with only one relevant document associated. (see fig. 5.4). The use of long queries can correspond in practice with, for example, a document recommendation system. Using short and "semantically tangential" queries (the second sentence from each simplewiki article), the performance of the bag-of-pairs falls below the one of bag-of-words, although it keeps a performance level

that is still suitable for practical purposes, when the measure uses queries with only one relevant document associated.

3. Using a weighted linear combination of the bag-of-pairs and the bag-of-roots scores (combination-of-bags), a modest but statistically consistent performance boost is achieved over the bag-of-words baseline. This effect is stronger under the configuration that is optimal for the standalone bag-of-pairs (see fig. 5.4), but it can also be appreciated in cases where the standalone bag-of-pairs method yields lower performance, e.g. using short queries. The optimal w depends on the configuration.

4. The coreference pronominal substitution produces a very small performance increase, not even statistically relevant in our tests. This may be due to the relatively high error rate of the coreference solver. Coreference resolution is in general a hard task, and despite being state of the art, the coreference resolution system we have used still produces a lot of mismatches compared to a human ([LPC+11] [bibb] indicate a performance of around 60% in a an unweighted average of three F-measures that take into account different ways of measuring precision and recall for a coreference solver).

5. In general, using queries with several relevant documents associated, affects more negatively the performance of the bag-of-pairs method. However, considering that some chunks may deal with subtopics that go off the main subject, we see more reliable the results obtained with whole articles. Regarding the fact that in our corpora this leads to having only one relevant document per query produces more noisy measures, we try to compensate it averaging over a higher amount of queries.

6. However, the many-relevant-articles set of queries, although less reliable, seems to confirm that the bag-of-pairs is stricter than the bag-of-words, that is, that some relevant documents are placed very close to the query because they satisfy certain special criteria, but the ones that don't are placed further away from the query than using the less strict method. This higher strictness would correspond with a noise source with higher kurtosis, if the obtained inter-document measure were interpreted as the sum of the underlying gold-standard relevance relation plus a certain noise. Our initial intuition was that bag-of-pairs would be stricter, because EP pairs are more specific than terms, as both terms in the pairs have to be the same in order for the pairs to be the same, but when this happens, it is more likely that they are associated to the same concepts. The confirmation comes from the relative approach to bag-of-words performance that bag-of-pairs exhibits in the precision/recall curves as the threshold is reduced (leftwards in figures 5.5 and 5.7).

7. As mentioned in the previous chapter, adding the rule sets 6b and 7 increases the performance of the system, despite the semantic irregularities introduced by these. This might be explained because different EP pairs holding different semantic relations will not be negative as long as the translations are consistent, and there is no substantial overlapping between them, so the detrimental effects are smaller than the advantage coming from the reduction of sparsity (we have more pairs per document now). For example, from "... at the 2003 MTV Video Music Awards" the following 'nn' dependencies are extracted: nn(Awards,MTV), nn(Awards,Video), nn(Awards,Music), producing the EP pairs (Awards,MTV), (Awards,Video), (Awards,Music). The third dependency is wrong: 'music' does not modify 'awards' but 'video', and thus 'awards' should have the property 'video' instead of 'music'. However, as long as this behaviour is consistent and other documents/queries with the same phrase are translated in the same way, and as long as other different phrases with different semantics do not produce also the EP pair '(awards,music)'. The fact that these awards are to some extent about music too is due to a transitivity effect that should not be relayed upon, as it will not hold in general. We don't find this argument applicable to the case of the semantic space, however, which justifies not using 6b and 7 for that case.

8. Clustered-bag-of-pairs and soft-bag-of-pairs do not improve the performance of bag-of-pairs, even for short queries, where a beneficial effect is more expected due to the extreme sparsity. In the case of soft-bag-of-pairs, the performance using the shaping function $f(x) = x^2$ is closer to the one of bag-of-pairs than using the identity, as was expected, but in both cases it is lower. However, we think the quality of the inter-term measure used is more to blame than the soft-bag-of-pairs method itself.

Summarizing, the only consistent improvement over the bag-of-words baseline comes from combination-of-bags. This improvement, which can be appreciated even in cases where the standalone bag-of-pairs gives a performance below the one from the baseline (bag-of-words) and the other linear component (bag-of-roots), might be explained as the result of w in equation 4.5 giving the optimal linear projection in a two-dimensional linear classifier. Deciding whether a given document is relevant or not for a given query can be seen as a classification problem with two features f_r and f_p given by the bag-of-roots and bag-of-pairs measures, consisting each of a hidden relevance variable $r \in 0, 1$ that depends on whether it is relevant or not, and an additive noise. That is: $f_r = r + n_r$,$f_p = r + n_p$. Even if a classifier using only f_p performs worse than one using f_r, because n_p has higher variance, or if both perform similarly, a Fisher linear classifier can yield a performance higher than the highest among both individual features, using an appropriate linear combination of both.

Additionally, the results should be contextualized with the following remarks.

a First, the error rate of the parser is significant, and will inevitably affect the results. The only way to test the performance of our system independently from the performance of the parser would be having a huge treebank of Stanford Dependencies built by human linguists, which would be very costly. In practice, this is a general issue that reduces any potential advantages of syntax over n-grams in NLP tasks. A similar point should be considered regarding the performance of coreference resolution, which was already discussed above.

b Also, the errors in the EPL-based semantic space inevitably affect the performance of the clustered-bag-of-pairs and soft-bag-of-pairs methods. The relaxation principle behind soft-bag-of-pairs seems sensible enough for such a strict method as the bag-of-pairs, as synonymy expansion is successfully used even for bag-of-words models. Therefore, we still thing that despite the lower performance obtained by soft-bag-of-pairs, a test with an inter-term similarity measure based on a hand-made thesaurus might yield better results than bag-of-pairs, and be on the right track to exceed the performance of the bag-of-words method without resorting to a combination with its scores.

c An extremely important factor that affects the experiments and contextualize the results is the corpus and the gold-standard relevance relation used between queries and documents. Our choice is based on queries (simplewiki articles) and documents (enwiki articles) having the same title, which produces a topic-based relation. It is possible that for most unordered sets of terms taken from a typical query, there is only one or very few possible ways that these can be syntactic and semantically arranged to refer to a sensible and typical topic. It is possible too that this effect is smaller for other kind of semantic features from text, such as sentiment or opinion.

6.2 Potential improvements and lines of research

In order to possibly improve the retrieval performance of EPL-based methods, other EPL versions could be tested, namely the ones with pseudo-terms, both for the semantic and document space. Our expectation is that in the case of the semantic space, pseudo-terms should be successfully placed according to their semantics, just as normative English terms. For instance "not.sleep" should be close to "awake", and "pat.giv" to "receiv". The main problem with the use of

pseudo-terms, and the main reason these extensions have not been carried out in this case, is the geometrical growth of the lexicon, which can easily become unmanageable.

Other improvements might stem from varying the contextualization factors pointed out in the previous section.

a Reducing the error rates of the parser and a coreference resolution system. However, this is an external factor, and we have already used a parser and a coreference resolution system with state-of-the-art performance.

b The inter-document measures that make use of an inter-term measure might yield better results if the latter were calculated over a hand-made thesaurus whose quality could be taken as a gold-standard, or an improved semantic space. The first possibility is more straightforward but less flexible, and the thesaurus should be adapted to the entity/property distinction. For the second possibility, options are including pseudo-terms, using a method more sophisticated than t_{EP} to make the inter-term measure statistically more reliable, and using distance and weightings other than cosine/IEF/IPF, for example the Jaccard distance and t-test weighting in [CM02a], which are claimed to give better results than plain cosine.

c Very importantly, other sets of queries and documents (or related documents in general) that were based in other criteria than topic alignment, could take more advantage of considering the internal structure of text. As mentioned before, an example of this might be sentiment or opinion alignment.

Other possible feature for the inter-document measure, which was partially implemented in our version but finally not included in the tests, is the use of differentiated entities within a particular document. That is, if the document includes references both to G. W Bush and H. W. Bush, instances of "Bush" referring to the former would be treated as different entities than those referring to the latter. This is equivalent to taking into account the integer markers in our example in figure 2.3, although in that particular example they didn't make a big difference. This could be expected to improve the semantic representation of some texts where this distinction is relevant, and could be implemented using the information from a coreference solver that determines not only the referent of pronouns but of any NP, as the Stanford coreference solver does. However, it was not included in the tests because through manual revision of some samples we noticed that the error rate of this non-pronominal coreference solving was very high, which is natural given the extreme difficulty of the task.

Appendix A

R-Prec plots

In this appendix we show the R-Prec results obtained from the configurations tested in chapter 5.3.

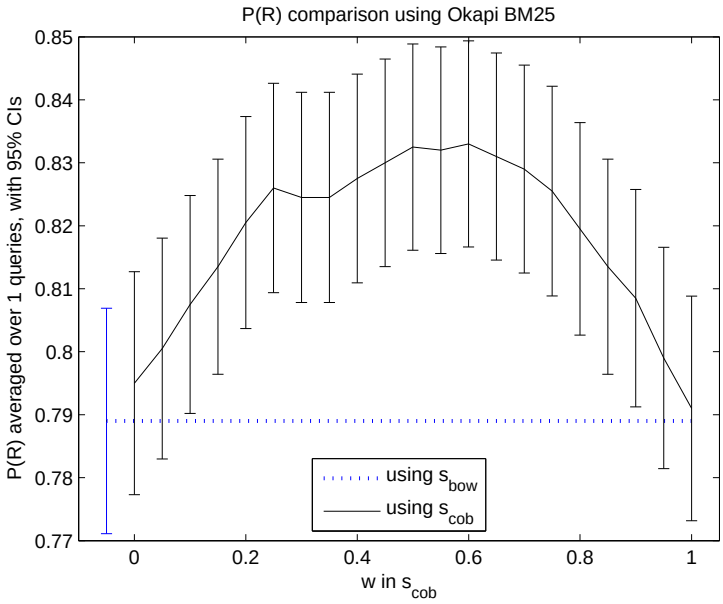

Figure A.1: R-Prec for configuration used in figure 5.4

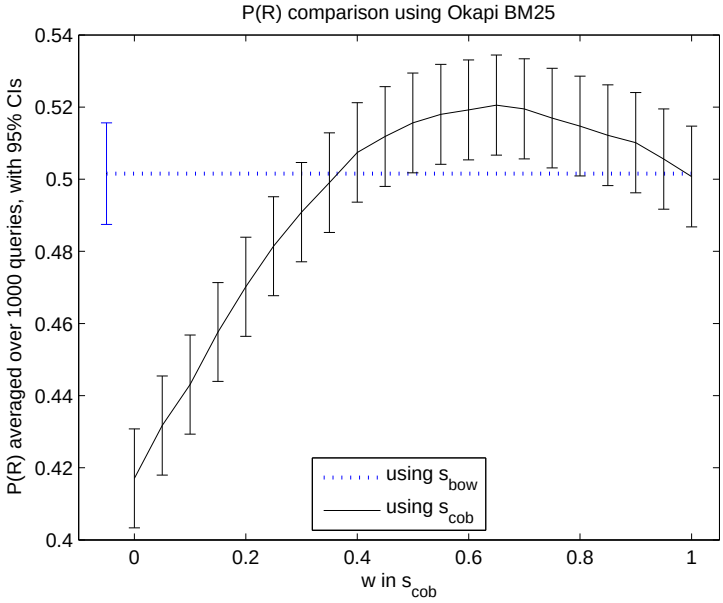

Figure A.2: R-Prec for configuration used in figure 5.5

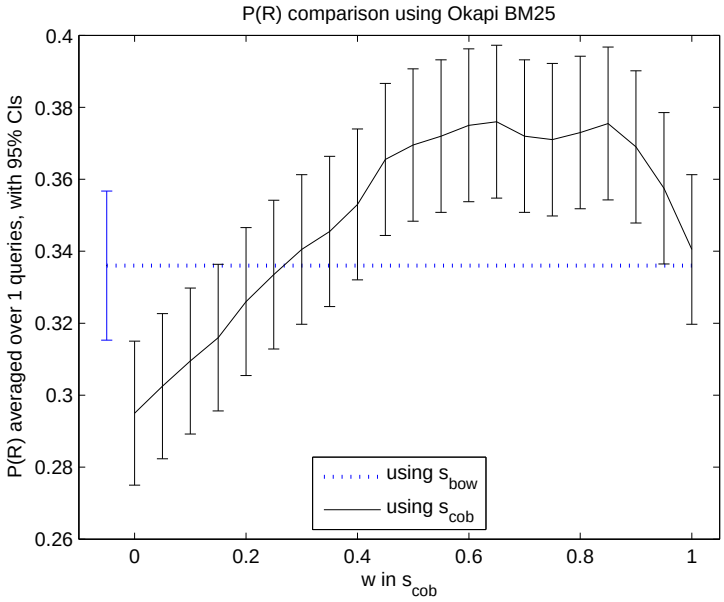

Figure A.3: R-Prec for configuration used in figure 5.6

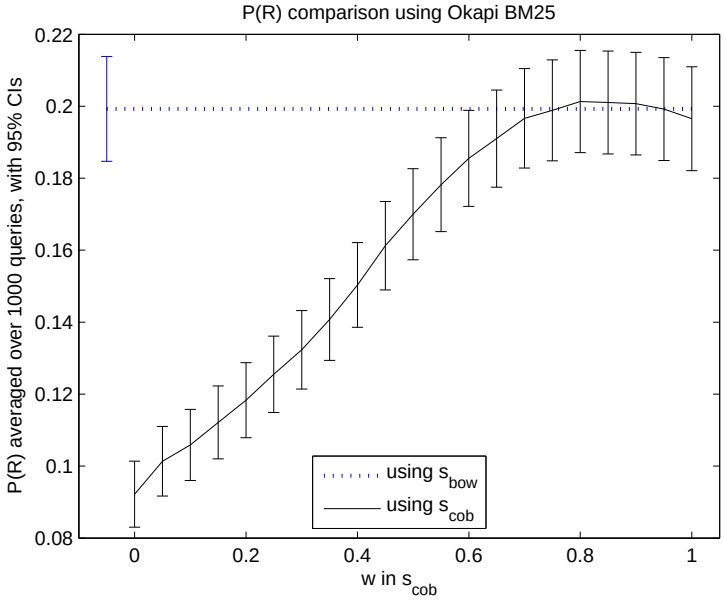

Figure A.4: R-Prec for configuration used in figure 5.7

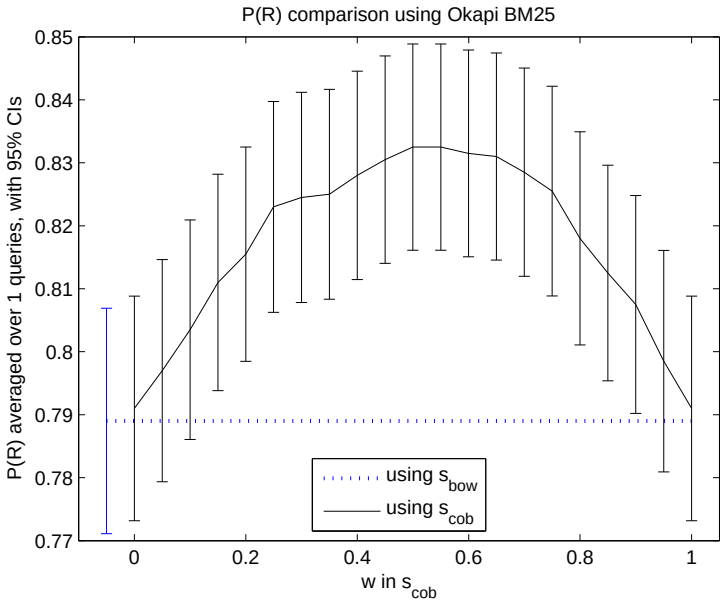

Figure A.5: R-Prec for configuration used in figure 5.8

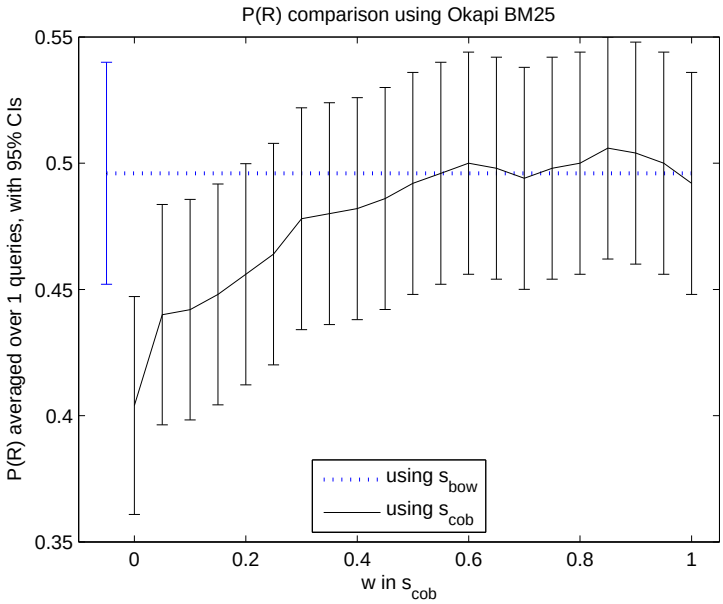

Figure A.6: R-Prec for configuration used in figure 5.9

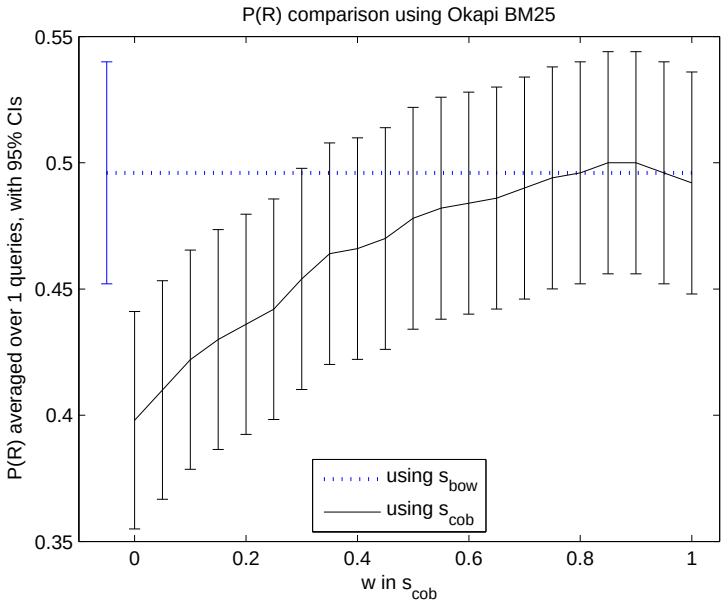

Figure A.7: R-Prec for configuration used in figure 5.10

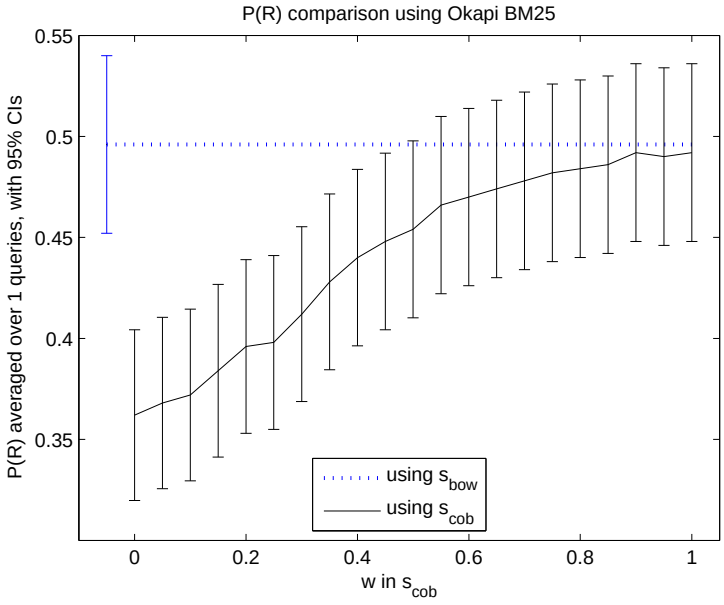

Figure A.8: R-Prec for configuration used in figure 5.11

Appendix B

Implementation details

This appendix discusses discusses further details regarding the implementation, including most practical issues. It is divided according to the different and relatively independent stages: preprocessing of the corpus in plain text format, parsing, and post-processing of the parsing from XML outputs to create the semantic and document spaces and testing the measures. The code of each section is publicly available at http://www.student.dtu.dk/~s094257/MScThesis/code/.

B.1 Corpus preprocessing

The version for the enwiki and the simplewiki dumps used are 20120307 and 20120331 respectively. The plain text has been extracted from the XML dumps with a customized version of the WP2TXT application [1], and subsequently processed, using Python, to meet some requirements before it is parsed by the CoreNLP library. These requirements are explained below.

While the CoreNLP parsing annotator runtime is linear on the text length, because sentences are parsed independently, the coreference resolution runtime is quadratic on the number of mentions, which in turn tends to be linear on the

[1] Available under the MIT license at http://wp2txt.rubyforge.org

text length [bibc]. Therefore, the articles have been restricted to a maximum of 1500 words, which is the point at which the coreference resolution tends to take as much time as the parser. For simplewiki, most articles already respect the size limit, and since they are going to be used as queries, the ones that exceed it have been discarded (see chapter 5). For enwiki, the articles longer than 1500 words have been split in chunks of approximately 1000 words. In order to reduce as much as possible the negative effects of this on the performance of the anaphoric coreference resolution, the divisions are set in points of the article that imply a partial change of topic: section divisions and paragraph divisions, in this order of preference. The algorithm is the following:

- The article is split in first-level sections.
- Consecutive sections are merged in a left-to right greedy fashion while they still satisfy the length limit.
- If some chunk is still larger than the limit, corresponding to a whole section that was larger, this is divided in paragraphs.
- Inside this section, consecutive paragraphs are merged in a left-to right greedy fashion while they still satisfy the length limit.
- If some paragraph is larger tan the limit (very unlikely), it is discarded.

Even though no quantitative evaluation has been performed, this strategy is expected to cause minimal reduction of performance for pronominal anaphoras, since after a new paragraph, the nominal referent is usually repeated by the writer, and this effect is even stronger for sections. The effect on non-pronominal coreference [2] is obvious, but this is of no relevance for the construction of the semantic space, because is only concerned about terms and doesn't distinguish between different word tokens with the same word type associated to different referents.

Meta-pages, such as "Wikipedia:Policies and guidelines" and the redirecting pages have been omitted. Also, inside every article, paragraphs with remnants of markup code have been filtered out, using manually tuned thresholds for the frequency of different sets of symbols that are uncommon, in different degrees, in English texts. This is not only to reduce noise in the results, but because some of this markup code produces out-of-memory problems later in CoreNLP (these problems are still present but in a much lower frequency, see section B.2).

The stop-words considered for the bag-of-words baseline method are in figure B.1.

[2] The one that would distinguish when instances of bush would refer to earlier mentions of George W. Bush or George H. W. Bush.

a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because, been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your

Figure B.1: The 119 terms used as stop-words.

B.2 Parsing

The corpora have been parsed with a modified CoreNLP library, using the HPC cluster at DTU with the Sun/Oracle Grid Engine. Since the parsing of each text file cannot be massively parallelized, several articles have been parsed in arrays of 60 parallel jobs.

The jobs themselves have been divided in batches of 50 text files each. The main reason for this is that the XML output is bigger than the text input by a factor of 100 (70 after modifying CoreNLP to omit some XML elements we didn't use), but highly redundant due to the XML tags. A zip compression reduces the factor 100 to only 7 (and 70 to 5). In order to keep bounded the space taken by unzipped xmls, these have to be regularly zipped, which can be done after each batch of XML files is produced.

Besides, with some malformed texts, the version of CoreNLP used (2012-03-09) produces out-of-memory errors that cannot be simply catched, but terminate the virtual machine abruptly, preventing the remaining text files in the filelist not to be parsed. One solution for this would be running a new invocation of CoreNLP for each text file in the job, but the overhead produced by the initialization of all the annotators at the beginning of each invocation would increase the running time several times. Therefore, we have included a routine in the calling jobs that creates a new filelist with the remaining text files every time CoreNLP exists with error, and runs CoreNLP again with this filelist.

The size of the jobs have been set to fulfill time and memory limitations imposed by the HPC hardware and queuing policies. A group of parallel jobs sent at a time to the HPC constitute what we call a dump (not to be confused with Wikimedia dumps, which comprise one entire Wikipedia each). Given the

configuration described above and the hardware at the time, dumps of 50 MB of plain text (2 MB each job) can be processed in 3 hours, so the entire enwiki could be parsed in 100 dumps of this size. To parse simplewiki, only a dump has been necessary.

B.3 EPL translation and vector operations

Matlab R2012a and R2010b have been used for decompressing and reading the XML outputs of CoreNLP both for enwiki and simplewiki common subsets, substituting pronouns using the coreference information, building an bag-of-words, bag-of-roots and different EPL representations of the documents and queries, performing different inter-documents measures, and comparing the obtained rankings with the gold standard relation defined by article titles, using the corresponding performance measures.

Matlab's provided datatype for sparse matrices uses Compressed Column Storage (CSS), which can be noticed by how the memory footprint of a sparse matrix approximates $(n_{\text{cols}} + c \cdot \text{nnz}) \cdot \text{wordsize}$ bytes, with c being a constant, $\text{wordsize} = 8$ in a x64 release, and nnz being the number of non-zero elements in the matrix. This is impractical for a vector-wise implementation of bag-of-pairs, because the dimensionality of the space is above n_{roots}^2, with $n_{\text{roots}} > 100000$ and even if EPL pairs are stored by rows both for documents and queries matrices, when both are EPL-pair-wise multiplied, one has to be transposed, Matlab attempts to calculate the intermediate transposed, and either runs out of memory or directly crashes (the latter happens with indexes higher than 2^{32} and may be a bug related to 32-bit integer representation). For this reason, ad-hoc representations have been implemented for the bag-of-pairs and soft-bag-of-pairs measures, and others derived from them. Namely, for the bag-of-pairs, entity and property indexes have been collapsed into a single index in the range $[1, n_{\text{roots}}^2]$, and using afterwards a sort of LIst of Lists (LIL) representation, with the pairs of each document being ordered by the collapsed index. This allows very efficient pair-wise multiplication of documents and queries, through an ordered-list intersection algorithm that runs in worst-case linear time respect to the number of pairs in the document (or the query in the uncommon case it has more pairs than the document). However, for soft-bag-of-pairs, as explained in chapter 4, a pairwise semantic comparison has to be made between entities in the query and the document, and the same between properties. The two resulting matrices, together with the matrix of values have to be element-wise multiplied and the result added up, which overall gives a cost proportional to $n_{\text{pairs-in-query}} \cdot n_{\text{pairs-in-document}}$. For this case, no optimization based on pre-ordering seems applicable.

Appendix C

Acronyms and abbreviations

bop	bag-of-pairs
bor	bag-of-roots
bow	bag-of-words
cbop	clustered-bag-of-pairs
cob	combination-of-bags
CSS	Compressed Column Storage
sbop	soft-bag-of-pairs
enwiki	English Wikipedia
EPL	Entity-Property Language
HPC	High Performance Cluster
IDF	Inverse Document Frequency
IEF	Inverse Entity Frequency
IPF	Inverse Property Frequency
IR	Information Retrieval
LIL	LIst of Lists
NLP	Natural Language Processing
NER	Named Entity Recognition
SD	Stanford Dependencies
simplewiki	Simple English Wikipedia
XML	eXtensible Markup Language

Bibliography

[biba] http://terrier.org/docs/v3.5/javadoc/org/terrier/matching/models/BM25.html. Accessed on 1 June, 2012.

[bibb] http://conll.cemantix.org/2011/. Accessed on 1 June, 2012.

[bibc] https://mailman.stanford.edu/pipermail/java-nlp-user/2011-March/000782.html. John Bauer. Accessed on 1 June, 2012.

[CM02a] J.R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9*, pages 59–66. Association for Computational Linguistics, 2002.

[CM02b] J.R. Curran and M. Moens. Scaling context space. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 231–238. Association for Computational Linguistics, 2002.

[DMM08] M.C. De Marneffe and C.D. Manning. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies_ manual. pdf*, 2008.

[DMMM06] M.C. De Marneffe, B. MacCartney, and C.D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[HHV10] T. Honkela, A. HyvÄrinen, and J.J. Väyrynen. Wordica—emergence of linguistic representations for words by independent component analysis. *Natural Language Engineering*, 16(03):277–308, 2010.

[LB01] J.P. Levy and J.A. Bullinaria. Learning lexical properties from word usage patterns: Which context words should be used. In *Connectionist Models of Learning, Development and Evolution: Proceedings of the Sixth Neural Computation and Psychology Workshop*, volume 273, page 282. Citeseer, 2001.

[Lin98] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics, 1998.

[LPC+11] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

[MRS08] C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[MSM99] C.D. Manning, H. Schütze, and MITCogNet. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.

[PL07] S. Padó and M. Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

[Sak07] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information processing & management*, 43(2):531–548, 2007.

[Sch98a] G. Schneider. A linguistic comparison of constituency, dependency and link grammar. *Master's thesis, University of Zurich*, 1998.

[Sch98b] H. Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998.

[SP95] H. Schütze and J.O. Pedersen. Information retrieval based on word senses. 1995.

[SP97] H. Schütze and J.O. Pedersen. A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing & Management*, 33(3):307–318, 1997.

[WHZ01] P. Wiemer-Hastings and I. Zipitria. Rules for syntax, vectors for semantics. In *Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society*, pages 1112–1117, 2001.