Using the X-IOTA System in Mono- and Bilingual Experiments at CLEF 2005

Loïc Maisonnasse[1], Gilles Sérasset[1], and Jean-Pierre Chevallet[2]

[1] Laboratoire CLIPS-IMAG, Grenoble France
loic.maisonnasse@imag.fr, gilles.serasset@imag.fr
[2] IPAL-CNRS, I2R A*STAR, National University of Singapore
viscjp@i2r.a-star.edu.sg

Abstract. This document describes the CLIPS experiments in the CLEF 2005 campaign. We used a surface-syntactic parser in order to extract new indexing terms. These terms are considered syntactic dependencies. Our goal was to evaluate their relevance for an information retrieval task. We used them in different forms in different information retrieval models, in particular in a language model. For the bilingual task, we tried two simple tests of Spanish and German to French retrieval; for the translation we used a lemmatizer and a dictionary.

1 Introduction

In the previous participation of the CLIPS laboratory in CLEF [1], we tested the use of surface-syntactic parsers in order to extract indexing terms. Last year, we only extracted simple indexing terms; this year we have tried to exploit the structure produced by the parser. We perforemd two separate evaluations; in the first one, we divided the structure into "complex descriptors", which contain part of the global structure. In the second one, we used a structure produced by the shallow parser, in a language model.

2 Sub-structure Training in the Monolingual Task

The shallow parser produces a structure, using only lemmas; we only use a part of the information produced . This year, we evaluated the relevance of the structural information produced by the parser. Two main types of parser are available; the dependency and the constituent. In our experiments we used a dependency parser; this kind of parser seems to be more appropriate for the information retrieval task [2] since it makes it possible to capture some sentence variation.

Different studies have already been made on the use of syntactic dependency structures. Some of these studies use dependency structure in order to extract phrases. For example, in [3], a closed structure is produced from a dependency tree for all sentences in a document. Some patterns are then applied on the structure for phrase extraction, and some selected phrases are then added to

C. Peters et al. (Eds.): CLEF 2005, LNCS 4022, pp. 69–78, 2006.
© Springer-Verlag Berlin Heidelberg 2006

other descriptors in the document index. Finally, the tf-idf weighting schema is adjusted in order to give a higher idf for the extracted phrase. In this way, a 20% gain over average precision is obtained. However, this gain cannot be directly linked to the use of a dependency structure since the structure is only used to detect the phrase.

On the presumption that converting the structures to phrases leads to the loss of information, other papers have tried to use the syntactic dependency structure directly. In [4], a dependency tree is extracted from Japanese sentences, mainly document titles. Matching between a query and documents is provided by a projection of the query tree onto the document trees. In addition, to provide a better matching, some pruning can be made on the tree. In [5], the COP parser (Constituent Object Parser) is used to extract dependency trees. In the query, the user has to select important terms and indicate dependencies between them. The query is then compared to the documents using different types of matching. The two papers cited provided just one unambiguous structure per sentence; [6] incorporates syntactic ambiguity into the extracted structure. The model proposed is applied to phrases; the similarity is provided by tree matching but the IR results are lower than the results obtained when only considering the phrases represented in the tree.

In our experiments, we consider an intermediary representation level. For this purpose, we use sub-structures composed of one dependency relation. With this representation, a sentence is considered as a set of sub-structures that we call dependencies. In our formalism, the sentence "the cat eats the mouse" is represented by the set: DET(the, cat), DET(the, mouse), SUBJ(cat, eat), VMOD(mouse, eat). Where "the" is the determiner of "cat", "cat" is the subject of "eat", etc.

2.1 Experimental Schema

For this experiment, we only used the French corpus. We experimented the use of dependency descriptors on this corpus. For this purpose, we use an experimental sequence, described in Figure 1.

First, the different documents of the collection are analysed with the French parser XIP (Xerox Incremental Parser) [7]. Two descriptors are extracted from these documents: the dependencies and the lemmas. In a first experiment, we considered these descriptors separately and created two indexes. One contains lemmas and the other dependencies. We queried these two indexes separately with dependencies and lemmas extracted from queries by the same parser. We compared the results obtained with the two descriptors for different weighting schemes. In a second experiment, we regrouped the two descriptors into a unique index and we evaluated results for different weighting schemes.

For training, we used the French corpus of CLEF 2003. In this corpus, there are 3 sets of documents. For each set, we selected the following fields: TITLE and TEXT for "le monde 94", TI KW LD TX ST for "sda 94" and "sda 95". For the queries, we selected the fields FR-title FR-descr Fr-narr.

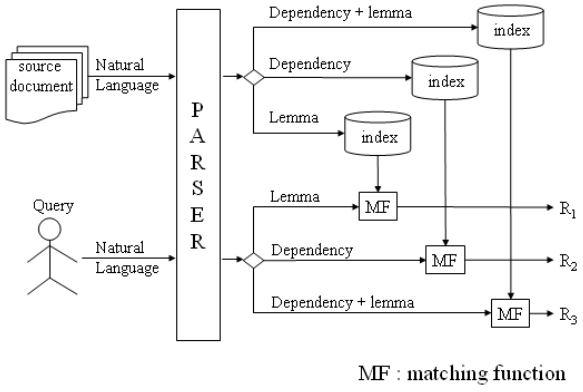

Fig. 1. Experimental procedure

2.2 Dependencies Versus Lemmas

We first compared results obtained using dependencies to results obtained with lemmas. In these experiments lemmas were used as the baseline as they have already shown their value in last year's CLIPS experiments [1]. After parsing the documents with XIP, we transformed the output into a common XML simplified format (shown below). From this XML format, on the one side we extracted the lemmas: for these descriptors, we only filtered nouns, proper nouns, verbs, adjectives and numbers.

XML simplified format for the sentence : "les manifestations contre le transport de déchets radioactifs par conteneurs." (Demonstrations against the transport of radioactive waste by containers)

```
<LUNIT>
<NODE num="2" tag="DET" lemma="le" ...>les</NODE>
<NODE num="3" tag="NOUN" lemma="manifestation" ... >
    manifestations</NODE>
<NODE num="5" tag="PREP" lemma="contre" ... >contre</NODE>
<NODE num="7" tag="DET" lemma="le" ... >le</NODE>
<NODE num="8" tag="NOUN" lemma="transport" ... >transport</NODE>
<NODE num="10" tag="PREP" lemma="de" ... >de</NODE>
<NODE num="12" tag="NOUN" lemma="dechet" ... >dchets</NODE>
<NODE num="14" tag="ADJ" lemma="radioactif" ... >
radioactifs</NODE>
<NODE num="16" tag="PREP" lemma="par" ... >par</NODE>
<NODE num="18" tag="NOUN" lemma="conteneur" ... >
conteneurs</NODE>
<NODE num="23" tag="SENT" lemma="." ... >.</NODE>
<DEP name="NMOD" ... w0="dechet" w1="radioactif"/>
<DEP name="NMOD" ...
w0="manifestation" w1="contre" w2="transport"/>
```

```
<DEP name="NMOD" ... w0="transport" w1="de" w2="d\'echet"/>
<DEP name="NMOD" ... w0="dechet" w1="par" w2="conteneur"/>
<DEP name="DETERM" ... w0="le" w1="manifestation"/>
<DEP name="DETERM" ... w0="le" w1="transport"/>
</LUNIT>
```

Table 1. Descriptor selected for the sentence: "les manifestations contre le transport de déchets radioactifs par conteneurs"

Selected lemmas	Selected Dependencies
manifestation	NMOD(déchet,radioactif)
transport	NMOD(manifestation,contre,transport)
déchet	NMOD(transport,de,déchet)
radioactif	NMOD(déchet,par,conteneur)
conteneur	DETERM(le,manifestation)
Allemagne	DETERM(le,transport)

On the other side, we extracted the dependencies (Table 1). As the number of dependencies can be very high, we queried each document set separately and then merged the results. We compared the IR results obtained with these two descriptors for different weighting schemes. We used the following weighting schemes on the document and on the query descriptors:

For the documents
nnn: Only the term frequency is used.
lnc: Use a log on term frequency and the cosine as the final normalization.
ltc: The classical tf*idf with a log on the term frequency.
nRn: Divergence from randomness

For the queries
nnn: Only the term frequency is used.
bnn: The binary model, 1 if terms are present, and 0 otherwise.
lnn: A log is used on the term frequency.
npn: Idf variant used by okapi.
ntn: classical idf.

For more details, see [1]. We first evaluated the c coefficient for the divergence from randomness weighting (nRn) on the document and with an nnn weighting on the queries. Results for the two descriptors are shown in Table 2 and 3. We then evaluated other weighting methods. The results are presented in Table 4.

Over all weighting schemes, dependency descriptors perform better than lemmas only for the nnn weighting. The divergence from randomness performs better than the other document's weighting for the two descriptors and the results are stable considering query weighting.

Table 2. Variation of c for nRn nnn (dependencies alone)

c	Average precision
2	25.53
3	25.50
4	25.83
4.25	25.93
4.5	26.01
4.75	26.00
5	25.88
5.5	25.84
6	25.84

Table 3. Variation of c for nRn nnn (lemmas alone)

c	Average precision
0	0.0152
0,5	0.4362
1	0.4647
1,75	0.4700
1,5	0.4703
2	0.4687
2,25	0.4728
2,5	0.4709
3	0.4577

Table 4. Lemmas or dependencies average precision

Document Weighting	Query Weighting									
	lemmas					dependencies				
	nnn	bnn	lnn	npn	ntn	nnn	bnn	lnn	npn	ntn
nnn	1,82	0,81	1,57	21,43	16,43	9,01	5,56	8,16	18,21	17,96
lnc	35,02	31,27	36,22	34,30	37,46	18,92	17,46	19,17	21,93	21,94
ltc	33,13	33,93	35,94	32,86	33,79	21,14	18,94	20,86	21,66	21,66
nRn	47,28	38,34	45,55	45,23	48,35	26,01	22,56	25,90	24,95	24,94

2.3 Lemmas and Dependencies

In our first experiment, we used dependencies and lemmas separately. In this second experiment we merged the two descriptors in one unique index and evaluated different weighting schemes for this index. Similarly to the previous experiment, we first evaluate divergence from randomness (Table 5) and the different weighting methods (Table 6).

The results obtained in this evaluation are better than those obtained with dependencies alone but they are lower than those obtain with lemmas. The reason is that the dependencies and the lemmas are considered as equivalent, whereas these two descriptors are clearly on two different levels as dependencies contain

Table 5. Variation of c for nRn nnn (lemmas and dependencies)

c	Average precision
0	0,0207
1	0,3798
1,5	0,3941
2	0,3947
2,25	0,3947
2,5	0,3934
3	0,3922

Table 6. Lemmas and dependencies average precision

Document Weighting	Query Weighting				
	nnn	bnn	lnn	npn	ntn
nnn	2.30	1.24	1.95	23.37	19.22
lnc	29.84	28.70	30.31	31.04	32.11
ltc	30.76	29.63	31.56	30.21	30.25
nRn	39.47	30.54	37.20	41.22	41.49

lemmas. This particular aspect was not taken into account in this experiment. Nevertheless, as we wanted to evaluate the use of dependencies, we submitted an official CLEF run with nRn nnn weighting with both dependencies and lemmas for the monolingual run and with the coefficient c at 2.25.

3 Language Models

In a second experiment, we integrated the syntactic structure in a language model. Some studies have already been made on the use of dependencies between terms in a language model in [8] [9]. These studies use statistical based methods in order to obtain a tree representation of a sentence; here we use a linguistically produced structure. In order to use a language model based on dependencies, from the previous XML simplified format, we have filtered only nouns, proper nouns, verbs, adjectives and numbers and the dependency that connects only these descriptors. For each sentence, we obtained a graph where the nodes are the significant elements of the sentence linked by dependencies (Figure 2). We used these graphs to apply a language model.

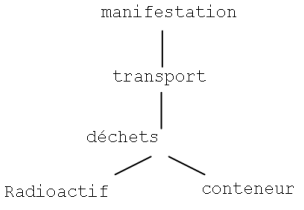

Fig. 2. Graph used by the langue model for the sentence: "les manifestations contre le transport de déchets radioactifs par conteneurs en Allemagne"

3.1 Our Language Model

The language model we used is a simplified version of the model proposed in [8]. This model assumes that the dependency structure on a sentence forms a undirected graph of term L and that the query generation is formulated as a two-stage process. At first a graph L is generated from a document following $P(L|D)$. The query is then generated following $P(Q|L, D)$; query terms are generated at this stage according to terms linked in L. Thus, in this model, the probability of the query $P(Q|D)$ over all possible graphs Ls is :

$$P(Q|D) = \sum_{L_s} P(Q, L|D) = \sum_{L_s} P(L|D)\, P(Q|L, D) \quad . \tag{1}$$

We assumed that the sum $\sum_{L_s} P(Q, L|D)$ over all the possible graphs L_S is dominated by a single graph L, which is the most probable graph. Here we consider that the most probable graph L is that extracted by our parser. We finally obtained:

$$P(Q|D) = log(P(L|D) + \sum_{i=1..m} P(q_i|D) + \sum_{(i,j)\in L} MI(q_i, q_j|L, D) \quad . \quad (2)$$

where: $MI(q_i, q_j|L, D) = log\left(\frac{P((q_i,q_j|L,D)}{P(q_i|D)P(q_j|D)}\right)$

$P((L|D)$. We estimate $P((L|D)$ as the probability that two terms are linked if they appear in the same sentences in the document. For this estimation, we made an interpolation of the document probability with the collection probability.

$$P(L|D) = \prod_{l\in L} P(L|D) \propto \prod_{(i,j)\in L} (1-\lambda_d)\frac{D_R(q_i,q_j)}{D(q_i,q_j)} + \lambda_d \frac{C_R(q_i,q_j)}{C(q_i,q_j)} \quad . \quad (3)$$

where l denotes a dependency between two terms

$D_R(q_i, q_j)$ denotes the number of time that q_i and q_j are linked in a sentence of the document

$D(q_i, q_j)$ denotes the number of time that q_i and q_j appear in the same sentence.

$C_R(q_i, q_j)$, $C(q_i, q_j)$ denotes the equivalent number but evaluated on the whole collection.

$P(qi|D)$. We estimate $P(qi|D)$ as the probability that a term appears in a document, and we made an interpolation on the collection.

$$P(q_i|D) = (1-\lambda_l)P(q_i|D) + \lambda_l P(q_i|C) \quad . \quad (4)$$

In the two last estimations, if a lemma or a dependency does not appear in the collection the probability is set to zero, consequently the whole probability will be set to zero. To avoid this, in the query we consider only the dependencies and the lemmas found in the whole collection.

$MI(qi, qj|L, D)$. We use the same estimation as the one used in [8].

3.2 Training

We applied this model on the CLEF 2003 collection. The results obtained are presented in Table 7 where we evaluate variations of the coefficients λ_l and λ_d.

We see that the results are better when the coefficient λ_l is around 0.3 and when the coefficient λ_d is high. Thus the results are better when the dependencies in the query are not taken into account. This may come from the use of simple estimations; better estimations of the probability may give better results. We submitted a run for this language model with the coefficient λ_l at 0.3 and the coefficient λ_d at 0.9999; the same experimental conditions were used.

Table 7. Average precision on variation of λ_l l and λ_d

λ_d λ_l	0.1	0.2	0.3	0.4	0.5	0.6
0.5	0.2749	0.2724	0.2697	0.2536	0.2495	0.2428
0.9999	0.2778	0.2951	0.2890	-	-	-

4 Bilingual

For the cross-language training, we performed two simple runs from German and Spanish to French. For these two runs, we used the three query fields : XX-title, XX-descr, XX-narr. In this training, query words are lemmatized and then translated using the web dictionary interglot[1].

For the lemmatization, we used TreeTagger[2] for the German queries and we used agme lemmatizer [10] for the Spanish queries. If there is an ambiguity with these lemmatizers, we keep all possible forms. We translate the lemmas with the dictionary and we keep all the translations found. Finally, we query the index of French lemmas with the divergence from randomness weighting.

For the CLEF 2003 test suite, we obtained an average precision of 0.0902 for the German queries and an average precision of 0.0799 for the Spanish queries.

5 Results

5.1 Monolingual

For this evaluation, we submitted three different runs. Two of these runs were based on dependencies with lemmas index with a weighting schema "nRn nnn" with the coefficient c at 2.25. The first FR0 used the fields FR-title FR-desc of the queries, the second FR1 used all the fields. The third run FR2 used the language model described in Section 3.1. We can see that as FR1 used the field FR-narr for the query the results are lower than the run FR0 which did not use this field. This may result from the fact that we did not use a program that processes the topics in order to remove irrelevant phrases as "Les documents pertinents doivent expliquer" (relevant documents must explain). We observe that the results obtained in CLEF 2005 are lower than those obtained for CLEF 2003, especially when we used the three query fields. In this case, the results for CLEF 2005 are more than two times lower than the results for CLEF 2003. This result may come from the fact that the narrative part of the queries seems to be shorter in CLEF 2005. Another difference could be that noticed between FR1 and FR2 as these two runs show a difference of about 10 points of precision for CLEF 2003 but are very close in CLEF 2005.

5.2 Bilingual

In this experiment, we submitted two runs for each source language. One of these two runs used the topic fields XX-title and XX-desc. The second also used the field XX-narr. The results obtained were lower that those obtained in training, they follow a decrease proportional to the monolingual. Thus this decrease appears to result from the low monolingual results.

[1] http://interglot.com/

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

Table 8. Monolingual results

	FR0	FR1	FR2
Average precision	21.56	14.11	13.07
Precision at 5 docs	38	36.40	30.40

Table 9. Bilingual results

	de-fr		es-fr	
	title +desc	title +desc +narr	title +desc	title +desc +narr
average precision	6.88	4.98	4.23	3.69
precision at 5 docs	17.20	12.80	10.80	11.60

6 Conclusion

For our participation in CLEF 2005 we evaluated the use of syntactic dependency structures extracted by a parser in an information retrieval task. In our first experiment, we tried to exploit the structure using descriptors that capture a part of the structure. In our second experiment, we directly exploited the structure extracted by the parser in a language model. The two experiments show that the structure is exploitable, but the results are still lower than those obtained using only lemmas with appropriate weightings.

As the syntactic structure has shown to be exploitable in IR, some improvements could be applied on this model. We used the XIP parser here, but this parser does not give information on the quality of the structure. Integrating this kind of information on the dependencies extracted could improve the IR results. Using a parser that extracts deeper syntactic dependencies may also give better results for the information retrieval task. Finally, our language model uses simple estimations, better estimations may improve the results.

Our conviction is that detailed syntactic information, which is already available using existing parsers, will improve results (especially, precision) in information retrieval tasks. However, such detailed information has to be combined with classical descriptors as, taken alone, it does not improve results. Obviously, we still have to find ways to combine the advantages of classical, raw descriptors with the added value of fine grain syntactic information in a single model. Independently of the task, we see that using the narrative part of the queries lowers our results. For our next participation, in order to improve our results, we will have to use a module that only selects the important part of the topic.

References

1. Chevallet, J.P., Sérasset, G.: Using surface-syntactic parser and deviation from randomness. In Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B., eds.: CLEF. Volume 3491 of Lecture Notes in Computer Science., Springer (2004) 38–49
2. Koster, C.H.A.: Head/modifier frames for information retrieval. In Gelbukh, A.F., ed.: CICLing. Volume 2945 of Lecture Notes in Computer Science., Springer (2004) 420–432

3. Strzalkowski, T., Stein, G.C., Wise, G.B., Carballo, J.P., Tapanainen, P., Jarvinen, T., Voutilainen, A., Karlgren, J.: Natural language information retrieval: TREC-7 report. In: Text REtrieval Conference. (1998) 164–173
4. Matsumura, A., Takasu, A., Adachi, J.: The effect of information retrieval method using dependency relationship between words. In: Proceedings of the RIAO 2000 Conference. (2000) 1043–1058
5. Metzler, D.P., Haas, S.W.: The constituent object parser: syntactic structure matching for information retrieval. ACM Trans. Inf. Syst. **7**(3) (1989) 292–316
6. Smeaton, A.F.: Using NLP or NLP resources for information retrieval tasks. In Strzalkowski, T., ed.: Natural language information retrieval. Kluwer Academic Publishers, Dordrecht, NL (1999) 99–111
7. A¿it-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond shallowness: incremental deep parsing. Nat. Lang. Eng. **8**(3) (2002) 121–144
8. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR, New York, NY, USA, ACM Press (2004) 170–177
9. Nallapati, R., Allan, J.: Capturing term dependencies using a language model based on sentence trees. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, New York, NY, USA, ACM Press (2002) 383–390
10. Gelbukh, A.F., Sidorov, G.: Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In Gelbukh, A.F., ed.: CICLing. Volume 2588 of Lecture Notes in Computer Science., Springer (2003) 215–220