Using Syntactic Dependency and Language Model X-IOTA IR System for CLIPS Mono and Bilingual Experiments in CLEF 2005

Loic Maisonnasse, Gilles Sérasset, Jean-Pierre Chevallet

Using Syntactic Dependency and Language Model
X-IOTA IR System Used for CLIPS Mono & Bilingual Experiments for CLEF 2005

Loïc Maisonnasse, Gilles Sérasset
Laboratoire CLIPS-IMAG, Grenoble France
Loic.maisonnasse@imag.fr

Jean-Pierre Chevallet
IPAL-CNRS, I2R A*STAR, National University of Singapore
viscjp@i2r.a-star.edu.sg

Abstract

This document describes the CLIPS experiments done for the CLEF 2005 campaign. We use surface-syntactic parser in order to extract new indexing terms. These terms are syntactic dependencies. Our goal is to evaluate their interest for an information retrieval task. We used them under different forms in different information retrieval models, particularly in a language model.

For the bilingual part we tried two simple tests on Spanish and German to French evaluation, for the translation we use a lemmatization and a dictionary.

1. Introduction

In the previous participation of the laboratory CLIPS at CLEF [1], we promote 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 extracted by the parser. By this, we have made two different evaluations; in the first one, we divided the structure extracted into complex descriptors, which contains a part of the global structure. In the second one, we use the syntactic dependencies between lemmas extracted by the shallow parser in a language model.

2. Sub-structure training on monolingual run

In this part, we evaluate the use of sub-structures as indexing terms. The shallow parser produces a structure, by using only the lemmas as we did last year, we only use a part of the extracted information. This year we evaluate the interest of the structural information produced by the parser. Different type of parser are available, in this paper we use a dependency parser as that kind of parser seems to be more appropriate for the information retrieval task [2].

Different works have already been made to use syntactic dependency structures. Some of those works use the dependency structure in order to extract phrases. For example, in [3] the author produces a structure close from a dependency tree for all documents sentences. After that, he applies some patterns on the structure for extracting phrases then some selected phrases are added to the others descriptors in the document index. At last, the tf-idf weighting schema is adjusted in order to give a higher idf for the extracted phrase. This way, a 20% gain on the average precision is provided on information retrieval results. Nevertheless, this gain cannot be directly linked with the use of a dependency structure.

On the presumption that converting the structures to phrases leads to lose information, others works have tried to use directly the syntactic dependency structure. In [4], a dependency tree is extracted from each Japanese sentence, mainly on the documents titles. The matching between a query and the documents is provided by a projection matching of the query tree on the documents trees. In addition, to provide, a better matching some cut 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 to indicate dependencies between them, the query is then compare to the documents by different types of matching. As the two preceding works provided only one unambiguous structure by sentence, in [6] the author incorporates syntactic ambiguity in the extracted structure. His model is apply on phrases, the similarity is provided by tree matching but the IR results are lower than the results obtained by considering only the phrases represented in the tree.

In our evaluation, we consider an intermediary representation level. For this purpose, we use sub-structures, which are composed by one dependency relation. By this representation a sentence is consider as a set of sub-structures that we call dependencies. In our formalism, the representation of 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. In this representation we lose only the information that there is two "the" in the sentence.

2.1. Experimentation schema

For this experiment, we only used the French corpus. On this corpus, we experiment the use of the dependency descriptor. For this purpose, we use an experimental sequence, described in the following schema:

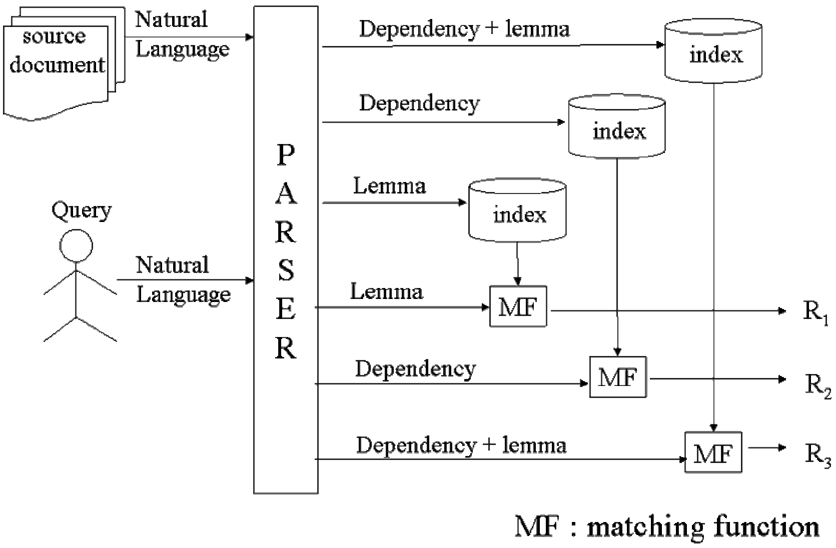

Figure 1: experimental procedure

First, the different documents of the collection are analysed with the French parser XIP (Xerox Incremental Parser) [7] from these documents two descriptors are extracted: the dependencies and the lemmas. In a first experiment, we consider these descriptors separately and create two indexes. One contains the lemmas and the other the dependencies. We query these two indexes separately with the dependencies and the lemmas extracted from the queries by the same parser. We compare the results obtain with the two descriptors for different weighting schemes. In a second experiment, we regroup the two descriptors in the same index and we evaluate the results for different weightings schemes.

For training, we have used the French corpus of CLEF 2003. In this corpus, there are 3 sets of documents. For each set, we have 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 have selected the fields FR-title FR-descr Fr-narr.

2.2. Dependencies versus lemmas

We have first compared the result obtain by using dependencies to the result obtain with lemmas. In these experimentation lemmas are used as baseline as they have already show they usability in the CLIPS experiments last year [1]. After parsing the documents with XIP, we have transformed the output into a common XML simplified format (Figure 2). From this XML, in one side we have extracted the lemmas: for these descriptors, we have filtered only nouns, proper nouns, verbs, adjectives and numbers.

```
<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="déchet" ... >déchets</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="déchet" w1="radioactif"/>
<DEP name="NMOD" ... w0="manifestation" w1="contre" w2="transport"/>
<DEP name="NMOD" ... w0="transport" w1="de" w2="déchet"/>
<DEP name="NMOD" ... w0="déchet" w1="par" w2="conteneur"/>
<DEP name="DETERM" ... w0="le" w1="manifestation"/>
<DEP name="DETERM" ... w0="le" w1="transport"/>
</LUNIT>
```

Figure 2: XML Information for the sentence : “les manifestations contre le transport de déchets radioactifs par conteneurs.” (Demonstrations against the transport of radioactive waste by containers)

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)

Figure 3: descriptor selected for the sentence: “les manifestations contre le transport de déchets radioactifs par conteneurs.”

On the other side, we extract the dependencies (Figure 3). As the number of dependencies can be very high, we query separately each documents set and then merge the results. We have compared IR results obtains with these two descriptors for different weighting schema. 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 is used as the final normalization.

ltc: The classical tf*idf with a log on the term frequency.

nRn: Derivation from randomness

For the queries

nnn: Only the term frequency is used.

bnn: The binary model, the terms presents are associated to the value 1, 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 have first evaluated the c coefficient for the derivation from randomness weighting (nRn) on the document and with an nnn weighting on the queries. Results for the two descriptors are shown on Table 1 and on Table 2. After that we have evaluated the others weighting methods, results are presented on Table 3.

c	Average precision
0	4.89
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
10	25.64

Table 1: Variation of c for nRn nnn (dependencies 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 2 variation of c for nRn nnn (lemmas alone)

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

Table 3: lemmas or dependencies average precision

Over all weighting, the dependencies descriptor performs better than the lemmas only for the nnn weighting. The derivation from randomness performs better than the others documents weighting for the two descriptors and the result are stable connecting to the query weighting.

2.3. Lemmas and dependencies

As in a first experiment we have used the dependencies and the lemmas separately. In this part, we merged the two descriptors in one index and evaluated the different weighting schemes for that index.

As for the preceding experimentation, we first evaluate the derivation from randomness (Table 4) and then we evaluate the different weighting methods (Table 5).

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 4: Variation of c for nRn nnn (lemmas and dependencies)

Doc	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

Table 5: lemmas and dependencies average precision

The results obtained in this evaluation are better than those obtain for the dependencies alone but they are lower than those obtain for the lemmas. The reason is that the dependencies and the lemmas are considered as equivalent descriptors when these two descriptors are clearly from two different levels as the dependencies contains the lemmas. This particularity was not taken in account in this experimentation. Nevertheless, as we want to evaluate the use of dependencies we have sent a run with nRn nnn weighting with both dependencies and lemmas for the monolingual run and with the coefficient c at 2.25.

3. Language model

In a second experimentation, we have integrated the structure extracted by XIP in a language model based information retrieval system. Some work have already been made in order to use dependencies between terms in a language model in [8][9]. These works used statistical based methods in order to get the tree representation of a sentence; here we used a linguistically extracted structure.

In order to use a language model based on dependencies, we have parsed the different documents sets. We have transformed the output into the XML simplified format. From this XML, we have filtered only nouns, proper nouns, verbs, adjectives and numbers and the dependency that connect only these descriptors. For each sentence, we obtain a graph where the nodes are the significant elements of the sentence linked by dependencies (Figure 4). We have used these graphs to apply a language model.

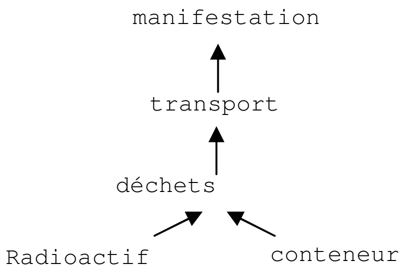

Figure 4: structure used by the langue model for the sentence: "les manifestations contre le transport de déchets radioactifs par conteneurs en Allemagne."

3.1. Language model

The language model we used is a simplified version of the model proposed in [8]**.** This model assumes that the query generation is formulated as a two-stage process. In a first time a linkage is generated from the document following the distribution *P(L|D)*. In a second time the query is generated following the distribution *P(Q|L,D)*, the query terms are generated at this stage according on the terms linked in *L*. So in this model the probability of the query *P(Q|D)* over all possible linkage *Ls* is :

$$P(Q|D) = \sum_{Ls} P(Q, L|D) = \sum_{Ls} P(L|D) P(Q|L, D)$$

The authors assume that the sum $\sum_{Ls} P(Q, L|D)$ over all the possible linkages *Ls* is dominated by a single linkage *L*, which is the most probable linkage. Here we consider that the most possible linkage *L* is those extract by our parser. The authors finally obtain:

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

Where $MI(q_i, q_j|L, D) = \log \frac{P(q_i, q_j|L, D)}{P(q_i|D)P(q_j|D)}$

Here we use simple estimation of the different parameters.

3.1.1. 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. On this estimation, we made an interpolation of the document probability with the collection probability. So we obtain the following formulation:

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

where *l* denotes a dependency between two terms

$D(q_i, q_j, R)$ 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.

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

3.1.2. P(q$_i$|D)

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

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

In the two last estimation, 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.

3.1.3. MI(qi, qj|L,D)

For this estimation, we use the same estimation as the one used in the article [8].

$$MI(q_i, q_j|L, D) = \log \frac{C(q_i, q_j, R)C(*,*,R)}{C(q_i,*,R)C(*,q_j,R)}$$

where *C(x,y,R)* denotes the count of link between the words x and y, the * symbolize all possible word.

3.2. Training

We apply this model on the Collection of CLEF 2003, the results obtain are presented in the following table where we evaluate the variation of the coefficients λ_l and λ_d.

λ_l / λ_d	Average precision
0,1/0,5	0.2749
0,2/0,5	0.2724
0,3/0,5	0.2697
0,4/0,5	0.2536
0,5/0,5	0.2495
0,6/0,5	0.2428
0,1/0,9999	0.2778
0,2/0,9999	0.2951
0,3/0,9999	0.2890

Table 6: varation of λ_l and λ_d

We see that the results are better when the coefficient λ_l is around 3 and when the coefficient λ_d is high. So the results are better when the dependencies in the query are not take in account. This thinks may comes from the fact that we use simple estimations; better estimation on the probability may give better results.

We submit 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 condition are used.

4. Bilingual

For the crosslingual evaluation, we have made two simple run from German and Spanish to French. For those two run, we use the three query fields : XX-title, XX-descr, XX-narr. And we query the French collection with the same fields as in the monolingual experiments. In this evaluation, the queries words are lemmatized and then we translate those lemma with the web dictionary interglot[1].

For the lemmatization, we use TreeTagger[2] for the German queries and we use agme lemmatizer [10] for the Spanish queries. On these lemmatizers if there is an ambiguity, we conserve all the possible forms. We translate the lemmas with the dictionary and we keep all the translations found. At last, we query the French lemmas index with the derivation from randomness weighting on the documents.

On clef 2003, we obtain 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 sent three different runs. Two of those runs were based on dependencies with lemmas index with a weighting schema 'nRn nnn' with the coefficient c at 2.25. The first of those two run 'FR0' use only the fields FR-title FR-descr of the queries, the second 'FR1' use the all fields. The last run 'FR2'submitted for the French monolingual task used the language model described in section 3. We can see that as 'FR1' use the field 'FR-narr' for the query his results are lover than the run 'FR0' which doesn't use this field. This artefact may comes from the fact that we have not used a program that process the topic in order to remove unrelevant phrase as 'Les documents pertinents doivent expliquer' (the relevant documents must explains). We observe that the results obtain on CLEF 2005 are lower than those obtain on CLEF 2003 specially when we used the three query fields the result for CLEF 2005 are more than two times lover 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 seen between 'FR1' and 'FR2' as these two runs show a difference of about 10 point of precision on CLEF 2003 but are very close on CLEF 2005.

[1] http://interglot.com/

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

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

Table 7 monolingual results

5.2. Bilingual

For this experiment, we send two runs for each source language. One of those two run used the topic field XX-title and XX-descr. The second use in addition of those two fields the field XX-narr.

	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 documents	17.20	12.80	10.80	11.60

Table 8: bilingual results

6. Conclusion

For our participation at clef 2005 we began to evaluate 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 by using descriptors that capture a part of the structure. In our second experiment, we exploited directly 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 here the XIP parser, 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 extract deeper syntactic dependencies may give better results for the information retrieval task. At last, our language model uses simple estimations, better estimation may improve the results.

Our conviction is that detailed syntactic information, which is already available using existing parsers, is to 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 show better 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 the next participation in order to improve our results we have to make an module that selects only the important part of the topic.

7. References

[1]. *Sérasset and Chevallet*, "Using surface-syntactic parser and Deviation from Randomness. X-IOTA IR system used for CLIPS Mono & Bilingual Experiments for CLEF 2004.". *Cross Language Evaluation Forum CLEF 2004,* pp.17-29, 2004

[2]. *Koster*, "Head/Modifier Frames for Information Retrieval". *CICLing 2004,* pp.420-432, 2004

[3]. *Matsumura, Takasu and Adachi*, "The effect of information retrieval method using dependency relationship between words.". *Proceedings of the RIAO 2000 Conference,* pp.1043-1058, 2000

[4]. *Matsumura, Takasu and Adachi*, "The effect of information retrieval method using dependency relationship between words.". *Proceedings of the RIAO 2000 Conference,* pp.1043-1058, 2000

[5]. *Metzler and Haas*, "The Constituent Object Parser: Syntactic Structure Matching for Information Retrieval.". *SIGIR'89, 12th International Conference on Research and Development in Information Retrieval,* SIGIR, pp.117-126, 1989

[6]. *Smeaton*, "Using NLP or NLP resources for information retrieval tasks.". Natural Language Information Retrieval, pp.99-111, 1999

[7]. *Ait-Mokhtar, Chanod and Roux*, " Robustness beyond shallowness: incremental deep parsing". Natural Language Engineering, pp. 121-144, 2002

[8]. *Gao et al.*, "Dependence language model for information retrieval.". *SIGIR-2004,* 2004

[9]. *Nallapati and Allan*, "Capturing term dependencies using a language model based on sentence trees.". *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management,* CIKM, pp.383-390, 2002

[10]. *Gelbukh and Sidorov*, "Approach to construction of automatic morphological analysis systems for inflective languages with little effort". *Computational Linguistics and Intelligent Text Processing (CICLing-2003),* pp.215–220, 2003