LexSchem: A Large Subcategorization Lexicon for French Verbs
Cédric Messiant, Thierry Poibeau

To cite this version:

HAL Id: hal-00321437
https://hal.archives-ouvertes.fr/hal-00321437
Submitted on 14 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
LexSchem: A Large Subcategorization Lexicon for French Verbs

Cédric Messiant and Thierry Poibeau

Laboratoire d’Informatique de Paris-Nord
CNRS UMR 7030 and Université Paris 13
99, avenue Jean-Baptiste Clément
F-93430 Villetaneuse France
firstname.lastname@lipn.univ-paris13.fr

Abstract
This paper presents LexSchem – the first large, fully automatically acquired subcategorization lexicon for French verbs. The lexicon includes subcategorization frame and frequency information for 3297 French verbs. When evaluated on a set of 20 test verbs against a gold standard dictionary, it shows 0.79 precision, 0.55 recall and 0.65 F-measure. We have made this resource freely available to the research community on the web.

1. Introduction
A lexicon is a key component of many current Natural Language Processing (NLP) systems. Hand-crafting lexical resources is difficult and extremely labour-intensive - particularly as NLP systems require statistical information about the behaviour of lexical items in data, and the statistical information changes from dataset to another. For this reason automatic acquisition of lexical resources from corpora has become increasingly popular.

One of the most useful lexical information for NLP is that related to the predicate-argument structure. Subcategorization frames (SCFs) of a predicate capture at the level of syntax the different combinations of arguments that each predicate can take. For example, in French, the verb “acheter” (to buy) subcategorizes for a single nominal phrase as well as for a nominal phrase followed by a prepositional phrase governed by the preposition “à”.

Subcategorization lexicons can benefit many NLP applications. For example, they can be used to enhance tasks such as parsing (Carroll et al., 1998; Arun and Keller, 2005) and semantic classification (Schulte im Walde and Keller, 2002) as well as applications such as information extraction (Surdeanu et al., 2003) and machine translation.

Several subcategorization lexicons are available for many languages, but most of them have been built manually. For French these include e.g. the large French dictionary “Le Lexique Grammaire” (Gross, 1975) and the more recent Leff (Sagot et al., 2006) and Dicovalence (http://bach.arts.kuleuven.be/dicovalence/) lexicons.

Some work has been conducted on automatic subcategorization acquisition, mostly on English (Brent, 1993; Manning, 1993; Briscoe and Carroll, 1997; Korhonen et al., 2006) but increasingly also on other languages, from which German is just one example (Schulte im Walde, 2002). This work has shown that although automatically built lexicons are not as accurate and detailed as manually built ones, they can be useful for real-world tasks. This is mostly because they provide what manually built resources don’t generally provide: statistical information about the likelihood of SCFs for individual verbs.

We have recently developed a system for automatic subcategorization acquisition for French which is capable of acquiring large scale lexicons from un-annotated corpus data (Messiant, 2008). To our knowledge, only one previously published system exists for SCF acquisition for French SCFs (Chesley and Salmon-Alt, 2006). However, no further work has been published since the initial experiment with this system, and the lexicon resulting from the initial experiment (which is limited to 104 verbs) is not publicly available.

Our new system is similar to the system developed in Cambridge (Briscoe and Carroll, 1997; Preiss et al., 2007) in that it extracts SCFs from data parsed using a shallow dependency parser (Bourigault et al., 2005) and is capable of identifying a large number of SCFs. However, unlike the Cambridge system (and most other systems which accept raw corpus data as input), it does not assume a list of predefined SCFs. Rather it learns the SCF types from data. This approach was adopted because at the time of development no comprehensive manually built inventory of French SCFs was available to us.

In this paper, we report work where we used this recent system to automatically acquire the first large subcategorization lexicon for French verbs. The resulting lexicon, LexSchem, is made freely available to the community under LGPL-LR (Lesser General Public License For Linguistic Resources) license.

We describe ASSCI, our SCF acquisition system, in section 2. LexSchem (the automatically acquired lexicon) is introduced and evaluated in section 3. We compare our work against previous work in section 4.

2. ASSCI: the subcategorization acquisition system
ASSCI takes raw corpus data as input. The data is first tagged and syntactically analysed. Then, our system produces a list of SCFs for each verb that occurred frequently enough in data (we have initially set the minimum limit to 200 corpus occurrences). ASSCI consists of three modules: a pattern extractor which extracts patterns for each target verb; a SCF builder which builds a list of candidate SCFs for the verb, and a SCF filter which filters out SCFs deemed incorrect. We introduce these modules briefly in the subse-
quent sections. For a more detailed description of ASSCI, see (Messiant, 2008).

2.1. Preprocessing: Morphosyntactic tagging and syntactic analysis

Our system first tags and lemmatizes corpus data using the Tree-Tagger and then parses it using Syntex (Bourigault et al., 2005). Syntex is a shallow parser for French. It uses a combination of heuristics and statistics to find dependency relations between tokens in a sentence. It is a relatively accurate parser, e.g. it obtained the best precision and F-measure for written French text in the recent EASY evaluation campaign1.

Our example below illustrates the dependency relations detected by Syntex (2) for the input sentence in (1):

(The drought came down on Sahel in 1972-1973.)

(2) DetFS|le|La|1|DET;2|
NomFS|sécheresse|sécheresse|2|S|SUJ;2,REF;3,PREP;5,PREP;8|
VCONS|abattre|abattit|4|SUJ;2,REF;3,PREP;5,PREP;8|
Prep|en|en|8|PREP;4|NOMPREP;9
NomMS|sahel|Sahel|7|NOMPREP;5|DET;6
Prep|sur|sur|5|PREP;4|NOMPREP;7
DetMS|le|le|6|DET;7|
NomMS|sèche|sèche|5|S|SUJ;4|DET;1
Prep|sur|sur|3|PREP;4|NOMPREP;7
NomFS|sècheresse|sècheresse|2|SUJ;2,REF;3,PREP;5,PREP;8
Prep|sur|sur|5|PREP;4|NOMPREP;7
DetMS|le|le|6|DET;7|
NomMS|sahel|Sahel|7|NOMPREP;5|DET;6
Prep|en|en|8|PREP;4|NOMPREP;9
NomXXDate|1972-1973|1972-1973|9|NOMPREP;8|
Typo|.|.|10||

Syntex does not make a distinction between arguments and adjuncts - rather, each dependency of a verb is attached to the verb.

2.2. Pattern extractor

The pattern extractor collects the dependencies found by the parser for each occurrence of a target verb. Some cases receive special treatment in this module. For example, if the reflexive pronoun “se” is one of the dependencies of a verb, the system considers this verb like a new one. In (1), the pattern will correspond to “s’abattre” and not to “abattre”. If a preposition is the head of one of the dependencies, the module explores the syntactic analysis to find if it is followed by a noun phrase (+SN) or an infinitive verb (+SINF).

(3) shows the output of the pattern extractor for the input in (1).

(3) VCONJS|s’abattre :
Prep+SN|sur|PREP..Prep+SN|en|PREP

2.3. SCF builder

The SCF builder extracts SCF candidates for each verb from the output of the pattern extractor and calculates the number of corpus occurrences for each SCF and verb combination. The syntactic constituents used for building the SCFs are the following:

1. SN for nominal phrases;
2. SINF for infinitive clauses;
3. SP[prep+SN] for prepositional phrases where the preposition is followed by a noun phrase. prep is the head preposition;
4. SP[prep+SINF] for prepositional phrases where the preposition is followed by an infinitive verb. prep is the head preposition;
5. SA for adjectival phrases;
6. COMPL for subordinate clauses.

When a verb has no dependency, its SCF is considered as INTRANS.

(4) shows the output of the SCF builder for (1).

(4) S’ABATTRE+s’abattre ;;; SP[|sur+SN|_SP[|en+SN]]

2.4. SCF filter

Each step of the process is fully automatic, so the output of the SCF builder is noisy due to tagging, parsing or other processing errors. It is also noisy because of the difficulty of the argument-adjunct distinction. The latter is difficult even for humans. Many criteria that exist for it are not usable for us because they either depend on lexical information which the parser cannot make use of (since our task is to acquire this information) or on semantic information which even the best parsers cannot yet learn reliably. Our approach is based on the assumption that true arguments tend to occur in argument positions more frequently than adjuncts. Thus many frequent SCFs in the system output are correct.

We therefore filter low frequency entries from the SCF builder output. We currently do this using the maximum likelihood estimates (Korhonen et al., 2000). This simple method involves calculating the relative frequency of each SCF (for a verb) and comparing it to an empirically determined threshold. The relative frequency of the SCF $i$ with the verb $j$ is calculated as follows:

$$rel.freq(scf_i, verb_j) = \frac{|scf_i, verb_j|}{|verb_j|}$$

$|scf_i, verb_j|$ is the number of occurrences of the SCF $i$ with the verb $j$ and $|verb_j|$ is the total number of occurrences of the verb $j$ in the corpus.

If, for example, the frequency of the SCF $SP[|sur+SN|]_SP[|en+SN]$ is less than the empirically defined threshold, the SCF is rejected by the filter. The MLE filter is not perfect because it is based on rejecting low frequency SCFs. Although relatively more low than high frequency SCFs are incorrect, sometimes rejected frames are correct. Our filter incorporates special
heuristics for cases where this assumption tends to generate too many errors. With prepositional SCFs involving one PP or more, the filter determines which one is the less frequent PP. It then re-assigns the associated frequency to the same SCF without this PP.

For example, $SP[\text{sur}+\text{SN}]_2SP[\text{en}+\text{SN}]$ could be split to 2 SCFs: $SP[\text{sur}+\text{SN}]$ and $SP[\text{en}+\text{SN}]$. In our example, $SP[\text{en}+\text{SN}]$ is the less frequent prepositional phrase and the final SCF for the sentence (1) is (5).

(5) $SP[\text{sur}+\text{SN}]$

Note that $SP[\text{en}+\text{SN}]$ is here an adjunct.

3. LexSchem

We used ASSCI to acquire LexSchem, the first fully automatically built large subcategorization lexicon for French verbs. We describe this work and the outcome in the subsequent sections.

3.1. Corpus

The automatic approach benefits from a large corpus. In addition, as we want our lexicon to be suitable for general use (not only for a particular domain use), the corpus needs to be heterogeneous enough to cover many domains and text types. We thus used ten years of the French newspaper Le Monde (two hundred millions words in total). Le Monde is one of the largest corpora for French and “clean” enough to be parsed easily and efficiently.

3.2. Description of the lexicon

Running ASSCI on this corpus data, we extracted 11,149 lexical entries in total for different verb and SCF combinations. The lexicon covers 3268 verb types (a verb and its reflexive form are counted as 2 different verbs) and 336 distinct SCFs.

Each entry has 7 fields:

- **NUM**: the number of the entry in the lexicon;
- **SUBCAT**: a summary of the target verb and SCF;
- **VERB**: the verb;
- **SCF**: the subcategorization frame;
- **COUNT**: the number of corpus occurrences found for the verb and SCF combination;
- **RELFREQ**: the relative frequency of the SCF with the verb;
- **EXAMPLES**: 5 corpus occurrences exemplifying this entry (the examples are provided in a separate file).

The following shows the LexSchem entry for the verb “s’abattre” with the SCF $SP[\text{sur}+\text{SN}]$.

```
:NUM: 05204
:SUBCAT: s’abattre \text{ : } SP[\text{sur}+\text{SN}]
:VERB: S’ABATTRE+\text{ }s’abattre
:SCF: $SP[\text{sur}+\text{SN}]$
:COUNT: 420
:RELFREQ: 0.882
:EXAMPLE: 25458;25459;25460;25461;25462
```

Two of the five corpus sentences exemplifying this entry are shown as follows (the syntactic analysis of Syntex is also available):

25458===Il montre la salle : On a fait croire aux gens que des hordes s’abattrent sur Paris .

25459===Dans ces conditions, sa réponse au problème politique corse est avant tout policière : avant 1981 , comme entre 1986 et 1988 , la répression s’abat sur les terroristes , souvent assimilés des délinquants de droit commun , et le pouvoir rejette toute idée de dialogue avec les "séparatistes " .

3.3. Evaluation

We evaluated LexSchem against a gold standard from a dictionary. Although this approach is not ideal (e.g. a dictionary may include SCFs not included in our data, and vice versa – see e.g. (Poibeau and Messiant, 2008) for discussion), it can provide a useful starting point. We chose a set of 20 verbs listed in Appendix to evaluate this resource. These verbs were chosen for their heterogeneity in terms of semantic and syntactic features, but also because of their varied frequency (200 to 100,000) in the corpus. We compared our lexicon against the Trésor de la Langue Française Informatisé (TLFI) - a freely available French lexicon containing verbal SCF information from a dictionary. We had to restrict out scope to 20 verbs because of problems in turning this resource into a gold standard.

We calculated type precision, type recall and F-measure against the gold standard, and obtained 0.79 precision, 0.55 recall and 0.65 F-measure. These results are shown in table 1, along with: 1) the results obtained with the only previously published work on automatic subcategorization acquisition (from raw corpus data) for French verbs (Chesley and Salmon-Alt, 2006), and 2) those reported with the previous Cambridge system when the system was used to acquire a large SCF lexicon for English with a baseline filtering technique comparable to the one employed in our work (VALEX sub-lexicon 2) (Korhonen et al., 2006). Due to the differences in the data, SCFs, and experimental setup, direct comparison of these results is unmeaningful. However, their relative similarity seems to suggest that LexSchem is a state-of-the-art lexicon.

The type precision and recall scores for each test verb are given in table 2.

3.4. The web distribution of LexSchem

LexSchem is freely available to the research community under the LGPL-LR (Lesser General Public License For Linguistic Resources) license.

http://www-lipn.univ-paris13.fr/lexschem.html. A web inter-

2See (Poibeau and Messiant, 2008) for details.

3http://infolingu.univ-mlv.fr/

DonneesLinguistiques/Lexiques-Grammaires/lgpllr.html
As mentioned earlier, the publicly available resource is Claire and Falk, 2005). Only part of this resource is currently in progress is aimed at addressing this problem (Gar- 

Informatis´e (TLFI) is derived from a syntax dictionary and

Figure 1: Comparison with recent work in French and English

Table 1: Comparison with recent work in French and En-

precision for verbs, adjectives and nouns. It is fairly comprehensive and offers also some ideas for further development of LexSchem. First, five different versions of the lexicon are provided in the web release at http://www.cl.cam.ac.uk/~alk23/subcat/lexicon.html. The idea is to provide different lexicons for the needs of different NLP tasks which vary in terms how accurate lexicons they require. For example, if the aim is to use SCF frequencies to aid parsing, it may be better to maximise the accuracy (rather than the cover-
5. Conclusion

This paper introduced LexSchem – the first fully automatically acquired large scale SCF lexicon for French verbs. It includes 11,149 lexical entries for 3268 French verbs. The lexicon is provided with a graphical interface and is made freely available to the community via a web page. Our evaluation with 20 verbs showed that the lexicon has state-of-the-art accuracy when compared with recent work using similar technology: 0.79 precision, 0.55 recall and 0.65 F-measure.

Future work will include improvement of the filtering module (e.g. experimenting with SCF-specific thresholds or smoothing using semantic back-off estimates), automatic acquisition of SCFs for other French word classes (e.g. nouns), and automatic classification of verbs using the SCFs as features (Levin, 1993; Schulte im Walde and Brew, 2002). Like mentioned above, we also plan to enhance the lexical entries of the lexicon. It would be useful to include in them information about noun and preposition classes and morpho-syntactic properties of the words included in SCFs.

Finally, as mentioned earlier, given different NLP applications have different requirements, it is worth building and releasing other versions of LexSchem.

Acknowledgements

This research was done as part of the ANR MDCO ‘CroTal’ project. It was supported by the British Council and the French Ministry of Foreign Affairs -funded ‘Alliance’ grant, by the EPSRC project ‘ACLEX’, and the Royal Society, UK. Cédric Messiant’s PhD is funded by a DGA/CNRS Grant.

6. References


Appendix — List of test verbs

<table>
<thead>
<tr>
<th>French Verb</th>
<th>English Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>aimer</td>
<td>like</td>
</tr>
<tr>
<td>apprendre</td>
<td>learn</td>
</tr>
<tr>
<td>continuer</td>
<td>continue</td>
</tr>
<tr>
<td>compter</td>
<td>count</td>
</tr>
<tr>
<td>croire</td>
<td>believe</td>
</tr>
<tr>
<td>jouer</td>
<td>play</td>
</tr>
<tr>
<td>offrir</td>
<td>offer</td>
</tr>
<tr>
<td>proposer</td>
<td>propose</td>
</tr>
<tr>
<td>refuser</td>
<td>refuse</td>
</tr>
<tr>
<td>rendre</td>
<td>render</td>
</tr>
<tr>
<td>s’abattre</td>
<td>fall</td>
</tr>
<tr>
<td>chercher</td>
<td>seek</td>
</tr>
<tr>
<td>concevoir</td>
<td>imagine</td>
</tr>
<tr>
<td>donner</td>
<td>give</td>
</tr>
<tr>
<td>monter</td>
<td>climb</td>
</tr>
<tr>
<td>ouvrir</td>
<td>open</td>
</tr>
<tr>
<td>possder</td>
<td>possess</td>
</tr>
</tbody>
</table>

(Appendix continues)