Automatic Collection of Related Terms in French

Xavier Robitaille, Satoshi Sato and Takehito Utsuro
Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University
xavier@pine.kuee.kyoto-u.ac.jp, {sato,utsuro}@i.kyoto-u.ac.jp

1 Introduction

It has long been known that a word's meaning is linked to the meaning of the words it co-occurs with. Firth (1957) epitomized this concept in his famous phrase: " You shall know a word by the company it keeps. " Likewise, technical terms (henceforth called *terms*) do not stand alone in their specific domain; rather, their meaning derives from their intrinsic relations with other terms. To understand these relations is to learn the domain.

This paper describes a web based method to extract French compound terms related to a given seed term. It is based on a method for the Japanese language proposed by Sato and Sasaki (2003), and has been adapted to the French language. Target applications include automatic or semi-automatic glossary compilation for a given domain. Also, if we worked out the correspondance between the French outputs and the Japanese outputs, we could create bilingual terminologies for specific topics.

In this paper, we start by giving a detailed overview of the system, focusing on the differences with the original Japanese system. Then, we describe our experiments and evaluate their results. Finally, we compare the outputs of both systems to assess the possibility of using them to automatically create bilingual terminologies.

2 System

We use a search engine to gather a set of related terms T for a given seed term s. Figure 1 shows the configuration of the system. We proceed in three steps: corpus compiling, automatic term recognition (ATR), and filtering.

2.1 Corpus compiling

We start by compiling a corpus C_s from web pages by selecting passages that describe s:

1. Web page collection
   We use Google to find relevant web pages by entering the following three queries: "s", "s est" (s is), and "s sont" (s are). We retreive the top 100 pages for each query, and parse those pages looking for hyperlinks whose anchor text contain s. If such pages exist, we also retreive them.
2. Sentence extraction
   From the retreived web pages, we remove html tags and other noisy text that doesn't resemble sentences. Then, we keep only properly structured sentences containing s, as well as the preceding and following sentences – that is, we use a window of three sentences around s.

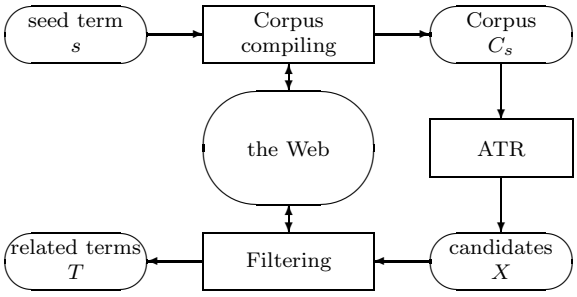

Figure 1: Overview of the system

2.2 Automatic term recognition

The next step is to extract candidate related terms from the corpus. Because the sentences composing the corpus are related to s, the same should be true for the terms they contain. We use the C-value method (Frantzi and Ananiadou, 2003), which extracts compound terms and ranks them according to their termhood. It consists of a linguistic part, followed by a statistical part.

1. Linguistic part
   The linguistic information consists of the part-of-speech tagging of the corpus and a linguistic filter constraining the type of

terms extracted. We base our filter on a morphosyntactic pattern for the French language proposed by Daille et al. (1994), which defines the structure of multi-word units (MWU) that are likely to be terms. Although their work focused on MWU limited to two main elements (nouns, adjectives, verbs or adverbs), we extend our filter to MWU of greater length. The pattern is defined as follows:

$$(Noun|Num)(Adj|PrepDet^{?}(Noun|Num))^{+}$$

2. Statistical part
For each compound that matches the linguistic pattern, we measure the termhood of the compounds – called C-value – by using statistical characteristics of the candidate string. It is given by:

$$C\text{-}value(a) = \begin{cases} \log_2 |a| f(a) \\ \qquad a \text{ is not nested}, \\ \log_2 |a| (f(a) - \frac{\sum_{b \in T_a} f(b)}{P(T_a)}) \\ \qquad otherwise \end{cases}$$

where a is the candidate string, $f(.)$ is its frequency of occurence in all the web pages retreived, T_a is the set of extracted candidate terms that contain a, and $P(T_a)$ is the number of these candidate terms.

2.3 Filtering

A filtering step is necessary because the set of candidates obtained by ATR is still noisy. The process is twofold. First, we use the C-value to remove unwanted compounds that match the linguistic pattern. Then, we use the hit count returned by search engines to determine whether the candidate is a valid related term.

2.3.1 C-value filter

Because we use a variable length pattern, if a long compound matches that pattern, all the shorter compounds it includes will also match. For example, consider the `Noun Prep Noun Prep Noun` structure in *systèmes à base de connaissances* (knowledge based system). The shorter candidate *système à base* (based system) also matches, although we would prefer to filter it out.

Fortunately, one of the strenghts of the C-value is the way it effectively handles nested multi-word terms. When we calculate the termhood of a string, we substract from its total frequency its frequency as a substring of longer candidate terms. In other words, a shorter compound that almost always appears nested in a longer compound will have a comparatively smaller C-value, even if its total frequency is higher than that of the longer compound. Therefore, we consider that a multi-word term is unwanted if its C-value is smaller than that of a longer candidate term in which it is nested.

2.3.2 $R_{\wedge/\vee}$ filter

Because we want to collect related terms, we need to verify that each output term x repects these two conditions: (1) the term is a *technical* term; (2) the term is *related* to the seed term s. To achieve this, we use the relation measure $R_{\wedge/\vee}$ (Sasaki and Sato, 2004). This measure has proved to be effective with the Japanese language and is expected to be language independant. It is given by:

$$R_{\wedge/\vee} = \frac{H(s \wedge x)}{H(s \vee x)}$$

where $H(.)$ is the hit count returned by the search engine, $H(s \wedge x)$ is the hit count of pages containing both s and x, and $H(s \vee x)$ is the hit count of pages containing s or x. The later can be calculated as follows:

$$H(s \vee x) = H(s) + H(x) - H(s \wedge x)$$

Candidates with high enough $R_{\wedge/\vee}$ measures are considered related terms of s. Table 1 shows the top 30 related terms extracted when using *linguistique informatique* (computational linguistics) in input.

3 Evaluation and results

3.1 Experiment

We input terms in our system and evaluated by hand whether each extracted candidate is a valid related term. Then, we evaluated the precision of the system by calculating the ratio of valid terms for different $R_{\wedge/\vee}$ thresholds. Table 2 gives the results for eight input terms and $R_{\wedge/\vee} > 0.01$, $R_{\wedge/\vee} > 0.02$ and $R_{\wedge/\vee} > 0.03$. Overall, these thresholds yield respective precisions of 65%, 79%, and 83%.

3.2 Adequacy of the C-value

As metioned above, we use the C-value to process nested multi-word terms. This proved to be quite effective: most correct nested terms were kept and most of the invalid ones were filtered out. For example, of the two shorter MWU

Table 1: First 30 candidates terms for *linguistique informatique*

$R_{\wedge/\vee}$	C-value	candidate term x	$H(x)$	$H(s \wedge x)$	$H(s/\vee x)$	related term?
0.1086	139	traitement automatique de la langue	1370	333	3067	✓
0.0910	544	traitement automatique	11400	1120	12310	
0.0806	210	traitement automatique du langage	1950	297	3683	✓
0.0747	137	industries de la langue	2000	280	3750	✓
0.0706	65	informatique linguistique	1670	244	3456	✓
0.0574	50	analyse linguistique	3110	279	4861	✓
0.0574	94	ingénierie linguistique	1970	217	3783	✓
0.0569	52	technologie du langage	477	135	2372	✓
0.0558	88	recherche en linguistique	1530	188	3372	✓
0.0555	77	langue naturelle	4130	324	5836	✓
0.0528	38	théories linguistiques	1660	185	3505	✓
0.0502	47	recherche en linguistique informatique	102	102	2030	✓
0.0496	170	linguistique appliquée	4470	307	6193	✓
0.0478	55	traitement du langage	2380	201	4209	✓
0.0457	20	sémantique des langues	489	110	2409	✓
0.0430	32	traitement de la langue	1440	143	3327	✓
0.0420	43	données linguistiques	1890	158	3762	✓
0.0405	101	analyse du discours	5040	275	6795	✓
0.0401	438	sciences du langage	14400	633	15797	✓
0.0379	25	résumé automatique	1040	112	2958	✓
0.0370	25	analyse automatique	2710	169	4571	
0.0363	40	dictionnaire électronique	2400	155	4275	✓
0.0348	43	linguistique de corpus	1150	107	3073	✓
0.0345	50	représentation des connaissances	3730	192	5568	✓
0.0336	30	langages formels	1970	130	3870	✓
0.0326	95	linguistique générale	5120	226	6924	✓
0.0297	50	option linguistique	949	86	2893	
0.0293	47	reconnaissance de la parole	3490	157	5363	✓
0.0284	218	traduction automatique	19300	591	20739	✓
0.0274	22	outils d' aide à la traduction	854	77	2807	✓

Table 2: Experimental Results

input	all	$X_{0.01}{}^{\dagger}$	$T_{0.01}{}^{\ddagger}$	prec. [%]	$X_{0.02}{}^{\dagger}$	$T_{0.02}{}^{\ddagger}$	prec. [%]	$X_{0.03}{}^{\dagger}$	$T_{0.03}{}^{\ddagger}$	prec. [%]
linguistique informatique	175	67	46	69	40	36	90	27	26	96
intelligence artificielle	148	72	55	76	34	31	91	12	11	92
traduction automatique	170	114	65	57	27	26	96	10	9	90
reconnaissance des formes	170	85	62	73	35	29	83	19	17	89
circuit logique	160	64	43	67	24	19	79	18	15	83
science cognitive	178	36	30	83	11	9	82	9	7	78
analyse vectorielle	173	91	35	38	50	26	52	24	18	75
reconnaissance vocale	152	67	43	64	24	15	62	11	7	64
all	1326	596	379	64	245	191	78	130	110	85

† X_{thr} : candidate terms with $R_{\wedge/\vee} > thr$

‡ T_{thr}: valid related terms with $R_{\wedge/\vee} > thr$

nested in *systèmes à base de connaissances* – *systèmes à base* (based system) and *base de connaissances* (knowledge base) – the filter removes only the former. This kind of behavior confirms that the C-value accurately measures the termhood of an MWU.

On the other hand, we note the C-value's inaptitude when it comes to the measuring the degree of relation between two terms. Take for instance the top three terms in table 1. The second term, *traitement automatique* (automatic treatment), is nested in the other two. From the point of view of the C-value, this is an indicator of independence, which is why it is given a high termhood (and isn't filtered out). However, from the point of view of our system, *traitement automatique* is a quite general term, which makes it inadequate as a related term of the seed *linguistique informatique* (computational linguistics). In other words, the C-value can favours general terms, precisely what we would like to avoid.

3.3 Adequacy of $R_{\wedge/\vee}$

A proof of the adequacy of $R_{\wedge/\vee}$ is the fact that, overall, precision increases with higher thresholds. This means that the measure successfully

Table 3: Correspondance with Japanese

input	$J_{0.01}$ †	$T_{0.01}$	$T'_{0.01}$ ‡	recall [%]	$T_{0.02}$	$T'_{0.01}$ ‡	recall [%]	$T_{0.03}$	$T'_{0.01}$ ‡	recall [%]
linguistique informatique	68	46	19	41	36	17	47	26	13	50
intelligence artificielle	111	55	23	42	31	18	58	11	10	91
traduction automatique	49	65	18	28	26	13	50	9	5	56
reconnaissance des formes	66	62	11	18	29	9	31	17	8	47
circuit logique	128	43	9	21	19	6	32	15	6	40
science cognitive	97	30	8	27	9	3	33	7	2	29
analyse vectorielle	58	35	10	29	26	10	38	18	7	39
reconnaissance vocale	90	43	14	33	15	7	47	7	5	71
all	667	379	112	30	191	83	43	110	56	51

† J_{thr} : Japanese candidate terms with $R_{\wedge/\vee} > thr$

‡ T'_{thr}: French related terms in T_{thr} for which a valid Japanese translation exists in J_{thr}

ranks the candidates accroding to their degree of relation with the seed term.

3.4 Observations

A striking characteristic of the French output is its semantic redundancy. In the candidates listed table 1, there are two equivalents of the term language processing: *traitement automatique de la langue* and *traitement automatique du langage*. Other examples include *reconnaissance des formes* and *reconnaissance de formes* for pattern recognition, or *sciences cognitives* and *sciences de la cognition* for cognitive sciences. This is due to the well known looseness of French compounds, and the difficulty to process them automatically (Gross, 1986) (Gross, 1990).

4 Correspondance with Japanese

Finally, we gave the same seed terms in input to both the French and Japanese systems, and looked for translation pairs in their outputs. More precisely, we translated all the valid French related terms above a certain $R_{\wedge/\vee}$ threshold, and tried to find them in the list of Japanese output candidates with $R_{\wedge/\vee} > 0.01$. The recall is given by the ratio of existing translation pair to the number of French related terms considered.

Table 3 shows the recall of translation pairs for French related terms with $R_{\wedge/\vee} > 0.01$, $R_{\wedge/\vee} > 0.02$, and $R_{\wedge/\vee} > 0.03$. Overall, these thresholds yield respectively 31%, 43%, and 57%. Again, increasing thresholds give better results. This experience confirms our hypothesis that translations of related terms in one language are likely to appear in the extracted terms of the another language.

References

J.R. Firth. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, pp. 1–32. Oxford: Oxford University Press.

S. Sato and Y. Sasaki. 2003. Automatic collection of related terms from the web. In *ACL-03 Companion Volume to the Proceedings of the Conference*, pp. 121–124.

佐々木靖弘、佐藤理史. 2003. 用語間の関連度を測る指標の提案. 言語処理学会第 10 回年次大会発表論文集, pp. 121–124.

B. Daille, E. Gaussier, and J.M. Lange. 1994. Towards Automatic Extraction of Monolingual and Bilingual Terminology. In *Proceedings of COLING 1994*, pp. 515–521.

K.T. Frantzi, and S. Ananiadou. 2003. The C-value/NC-value domain-independent method for multi-word term extraction. In *Journal of Natural Language Processing*, 6(3), pp. 145–179

M. Gross. 1986. Lexicon-Grammar. The Representation of Compound Words. In *Proceedings of COLING 1986*, pp. 1–6.

M. Gross. 1990. Les mots composés. In *Modèles linguistiques*, 12 (1), pp. 47–63.