Automatic Term Extraction Using Log-Likelihood Based Comparison with General Reference Corpus

Alexander Gelbukh[1], Grigori Sidorov[1],
Eduardo Lavin-Villa[1], and Liliana Chanona-Hernandez[2]

[1] Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, Zacatenco,
Mexico DF, 07738, Mexico
[2] Engineering faculty (ESIME),
National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, Zacatenco,
Mexico DF, 07738, Mexico
www.gelbukh.com, www.g-sidorov.org

Abstract. In the paper we present a method that allows an extraction of single-word terms for a specific domain. At the next stage these terms can be used as candidates for multi-word term extraction. The proposed method is based on comparison with general reference corpus using log-likelihood similarity. We also perform clustering of the extracted terms using k-means algorithm and cosine similarity measure. We made experiments using texts of the domain of computer science. The obtained term list is analyzed in detail.

Keywords: Single-word term extraction, log-likelihood, reference corpus, term clustering.

1 Introduction

Automatic term extraction is an important task in the field of natural language processing [1]. Even preliminary term extraction with certain degree of errors for further manual processing is very useful. Extracted terms can be used, for example, in ontology construction, information retrieval, etc.

Manual term extraction is possible but it relies on human knowledge of an expert, so, this process is expensive and very slow. Another important consideration is that the extracted terms would be subjective [9], i.e. the experts would have different opinions, while the automatic processing is more objective though it depends on the availability of the corpus data.

Traditionally, the investigations on term extraction are focused on extraction of multi-word terms. POS tagging and various parsers, as well as statistical methods are used in this task [6]. In this case, the main purpose of the statistical methods is the evaluation of the strength of connection between words in multi-word terms.

C.J. Hopfe et al. (Eds.): NLDB 2010, LNCS 6177, pp. 248–255, 2010.

In our opinion, this task should be separated into two steps. At the first step, we detect most probable single words that are candidates for being terms in a specific domain. At the second step, we can apply techniques for multi-word term extraction for obtained single-word terms.

In this paper, we present a method that corresponds to the first step of term extraction (single-word terms extraction) and corresponding experiments for Spanish language. We also perform automatic clustering of terms.

Further in the paper we first describe the method, then present detailed results of our experiments and discuss them, and finally conclusions are drawn.

2 Description of the Term Extraction Method

The proposed method is a modified version of the method presented in [8] for Chinese language. The input data are texts from a specific domain: in our experiments we used texts of the domain of computer science. Also, some general reference corpus should be used for comparison. It is expected that the reference corpus is rather big because otherwise we will not be able to filter out general words of the domain corpus.

The general idea is related to comparison of weighted frequencies in the two corpora: if a word appears much more frequently in the domain corpus, it is a probable term. Note that in [8] it is stated that log-likelihood comparison gives better results than more traditional *tf-idf* based comparison.

There are two main stages during the whole processing: preprocessing and proper term extraction using log-likelihood.

We modified the method [8] in the following aspects: it is applied to Indo-European language (Spanish) with corresponding changes in preprocessing; we do not use any enrichment with additional resources (for example, WordNet) that is important part of the original method; we changed the formula for calculation of the log-likelihood similarity: instead of using more traditional log-likelihood test, we calculate the log-likelihood based distance [7], [2]. Note that this distance measure was precisely developed for comparison of corpora. We also was obliged to add an additional step in the calculations that distinguishes the domain terms as compared to possible words from the general corpus with the same properties, see Formula 4.

Several operations are performed at the preprocessing step. First we tokenize the texts. Then all words are changed to the unified register. We ignore punctuation marks, special symbols, and numbers; all words are lemmatized using freely available lemmatizer for Spanish developed in our laboratory. We filter out auxiliary words: prepositions, articles, auxiliary words, etc.

For calculation of weights of words we used the following formula [7].

$$G = 2 * \left(\begin{array}{l} \left(freq_{domain} * \log\left(\frac{freq_{domain}}{freq\_Expected_{domain}} \right) \right) + \\ \left(freq_{general} * \log\left(\frac{freq_{general}}{freq\_Expected_{general}} \right) \right) \end{array} \right) \quad (1)$$

where $freq_{domain}$ and $freq_{general}$. are real frequencies in the domain corpus and in the reference corpus.

$freq\_Expected_{domain}$ and $freq\_Expected_{general}$ are expected frequencies in the domain corpus and in the reference corpus. They are calculated according to the following formulae:

$$freq\_Expected_{domain} = size_{domain} * \frac{freq_{domain} + freq_{general}}{size_{domain} + size_{general}} \quad (2)$$

$$freq\_Expected_{general} = size_{general} * \frac{freq_{domain} + freq_{general}}{size_{domain} + size_{general}} \quad (3)$$

As the result of application of this formula, all words in the domain corpus are assigned weights.

Another important step in the algorithm consists in the following. Note that Formula 1 cannot distinguish to which of the two corpora the term belongs, i.e. Formula 1 is symmetrical for both corpus. We should somehow correct this situation because we are searching terms specifically in the domain corpus, and not all words with similar properties. So, we add an additional condition: we take into account only terms that satisfy Formula 4, i.e. their relative frequency is bigger in the domain corpus than in the reference corpus. If this condition is false, then we discard the word as a possible term: in our case, we multiply its weight by -1.

$$\frac{freq_{domain}}{size_{domain}} \rangle \frac{freq_{general}}{size_{general}} \quad (4)$$

After application of Formula 4, some words of the domain corpus will be discarded as possible terms: their weight will be negative. See an example given in Table 1.

Table 1. Examples of the calculated data

Word	$freq_{domain}$	$freq_{general}$	$freq\_Expected_{domain}$	$freq\_Expected_{general}$	G
...					
socket	1	0	0.010	0.989	9.153
sofisticado (sofisticated)	5	169	1.789	172.210	3.912
soft	1	12	0.133	12.866	2.351
software	430	831	12.971	1248.028	2334.961
sofware[1]	2	2	0.041	3.958	12.803
sol (sun)	2	933	9.618	925.381	-9.016
solamente (just)	20	1714	17.837	1716.162	0.254
...					

[1] This is a spelling error in the corpus, the correct form is *software*.

It can be seen that the word *software* has very high weight. At the same time, though the word *sol* (*sun*) also has rather high weight, it is discarded because its relative frequency in the reference corpus is greater than in the domain corpus.

After application of this processing we have a list of words from the domain corpus ordered by their weight. Still, the question remains what is the value of the threshold for selection of the upper part of this list. For the moment we use the empirical value for this threshold, see below.

3 Experiments and Discussion

In our experiments we used the following data (in Spanish). We used issues of the *Excelsior* newspaper (Mexico, 1990s) as a general reference corpus, total 1,365,991 running words. We used texts related to computer science loaded from Wikipedia as a domain corpus, for example, articles about *informatics, software, programming*, etc. Totally we used 26 articles that contain 44,495 words.

After several experiments, we decided empirically to use threshold of 270 elements for the selection of the terms with the highest weight. In the paper [8], threshold of 216 terms was selected, but they also used additional analysis of relations in Chinese analogue of WordNet.

After extraction of terms and selection of the set of high scored terms, we cluster them using standard *k-means* algorithm. Note that *k-means* algorithm needs a manual selection of number of classes for clustering. For calculating of the similarity during clustering we used the standard cosine measure calculated over *tf-idf* values. We used an empirical threshold of 19 classes for this algorithm.

An interesting question arises if verbs should be part of the list of terms because in technical writing verbs are often lexical functions to the corresponding nouns. For example, for a noun *program* the lexical function $Oper_1$ will be *design (a program)* or *develop (a program),* see [5]. Similar question arises for nouns derived from the corresponding verbs, e.g., *development*. For the moment, we decided to exclude verbs for evaluation of results, but leave the derived nouns.

Results of one of the experiments are presented in **Table 2**. The words are clustered using *k-means* algorithm. The first word in the cell is the centre of the class. Some words were extracted in English, like, *for, to, DAQ,* etc., i.e. in the form they were represented in the source texts. The extracted words belong to various fields of computer science: programming, bioinformatics, electronics, etc.

In the table, we stroke out the words that are clearly errors of the term detection algorithm. Some of these errors can be easily corrected, like, *to* or *etc.* We also underlined verbs, which we do not take into account.

If we have a look at the obtained list, it can be easily seen that it has many words that are terms of science in general, like *analysis, model, science, theory*, etc. We marked these words with italic. It is not clear if these words are errors of the method or not. Since we make comparison with general reference corpus, we do not have the possibility to distinguish them from more specific terms. On the other hand, it seems very plausible that we can detect them if we also make a comparison with a domain corpus of other scientific field.

Table 2. Obtained classes of extracted single word terms

Classes of detected terms (Spanish)	Classes of detected terms (translated)
algoritmo, for, *implementación*, array, implementar, árbol	algorithm, for, *implementation*, array, implement, tree (search)
analógica, voltaje, binario	analog, voltage, binary
as, if, int, integer, pseudocódigo, return, vtemp, *diagrama*, *descripción*, Turing, end	as, if, int, integer (number), pseudocode, return, vtemp, *diagram, description,* Turing, end
b2b, *business*, hosting, cliente, servidor, internet, ~~to~~, electrónico, consistir	B2B, *business*, hosting, client, server, Internet, ~~to~~, electronic, consist
biología, bioinformática, ADN, alineamiento, clustalw, fago, gen, genoma, genomas, genome, genómica, génica, homología, *human*, microarrays, *modelado*, nucleótidos, *predicción*, proteína, proteína-proteína, sanger, secuenciación, evolutivo, *secuencia*, *biológico*, computacional, protocolo, *variedad*, *análisis, técnica, estructura, interacción*, completar, *montaje, herramienta,* ~~menudo~~, usar, talar, software, visualizar, *cuantificación*, *modelo*, automatizar, búsqueda	*biology*, bioinformatics, DNA, alignment, ClustalW, fag, gene, genome, genomes, genomics, genetic, homology, *human*, microarrays, *modeling*, nucleotides, *prediction*, protein, protein- protein, sanger, sequencing, evolutionary, *sequence*, *biological*, computational, protocol, *variety*, *analysis, technical, structure, interaction* complement, assembly, tool, ~~often~~, use, fell, software, visualize, *quantification, model*, automate, search
componente, transistor, tubo, funcionar, conexión, dispositivo, ~~etc~~, *tecnología*, digitales, microprocesadores, *velocidad, lógica*, soler, altavoz	component, transistor, tube, function, connection, device, ~~etc~~., *technology*, digital, microprocessor, *speed*, *logic*, happens, speaker
computación, *ciencia, constable, científica,* cómputo, *disciplina, matemática*, ~~usualmente~~, *teoría*, computacionales, *ingeniería*, estudiar, artificial, *matemático*, informática, paralelo, programación	computer, science, *constable*, *scientific*, computing, *discipline*, *mathematics*, ~~usually~~, *theory*, computing, *engineering*, study, artificial, *mathematical*, informatics, parallel, programming
conjunto, *notación, problema*, finito, binaria, complejidad, np, np-completo, *número, tamaño, elemento*, coste, lineal, ~~comúnmente~~, ~~montículo~~	set, *notation, problem*, finite, binary, complexity, np, np-complete, *number, size, item,* cost, linear, ~~commonly~~, ~~mound~~
código, compilador, compiladores, lenguaje, máquina, programa, ~~compuesto~~	code, compiler, compilers, language, machine, program, ~~consisting~~

Table 2. *(continued)*

descifrar, criptografía, cifrar, *método*, texto, denominar	decode, cryptography, encrypt, *method*, text, name
dimensión, cubo, *espacial*, almacén, marts, metadato, middleware, warehouse, data, olap, tabla, operacional, variable, *definición*, especificar, usuario, poseer, almacenar, dato, colección, arquitectura, registro	*dimension*, cube, *space*, warehouse, marts, metadata, middleware, warehouse, data, olap, chart, operational, variable, *definition*, specify, user, possess, store, data, collection, architecture, register
diseñar, diseñador, objeto, funcional, procesar, proceso	*design, designer*, object, functional, process, process
formato, avi, compresión, *especificación*, formatos, mov, archivar, vídeo, audio, archivo, informático, codificar, *estándar*	format, avi, compression, *specification* formats, mov, archive, video, audio, file, computer, code, *standard*
potencia, válvula, analógicos, semiconductor, corriente, alternar, analizador, electrónica, conmutación, eléctrico, sonido, pila, supercomputadoras	power, valve, analog, semiconductor, power, switch, analyzer, electronic, switching, electrical, audio, battery, supercomputers
red, *principal*, *artículo*, permitir, utilizar, ~~varío~~, aplicación, información, ~~través~~, ~~tipo~~, *sistema*, ~~ejemplo~~, característica, interfaz, *forma*, gestión, operativo, acceder, ~~diferente~~, base, contener, operación, función, clasificar, ordenador, ejecutar, programador, cálculo, modelar, relacionales, interfaces, objeto, relacional	network, *main, article*, allow, use, ~~various~~, application, information, ~~through~~, ~~type~~, *system*, ~~example~~, feature, interface, *form*, management, operating, access, ~~different~~, base, contain, operation, function, classify, computer, execute, programmer, calculation, model, relational, interfaces, object, relational
~~rápido~~, acceso, ~~sencillo~~, soportar, web, ~~específico~~, ~~central~~, fiabilidad, paralelismo	~~fast~~, access, ~~easy~~, support, web, ~~specific~~, ~~central~~, reliability, parallelism
señal, transductores, transductor, impedancia, filtrar, conversión, acondicionamiento, convertidor, daq, *adquisición*, analógico, conectar, adaptación, frecuencia, medir, tensión, sensores, digital, cable, control, ~~física~~, entrada, medición, ~~físico~~, salida, ~~normalmente~~, bus, dato	signal, transducers, transducer, impedance, filter, converting, packaging, converter, daq, *acquisition*, analogue, connect, adapt, frequency, measure, voltage, sensors, digital, cable, control, ~~physics~~, input, measurement, ~~physical~~, output, ~~normally~~, bus, data
térmico, ci, cápsula, integration, scale, chip, circuito, chips, integrar, híbrido, silicio, reproductor, amplificador, ~~fabricación~~	heat, ci, capsule, integration, scale, chip, circuit, chips, integrated, hybrid, silicon, player, amplifier, ~~manufacturing~~

Some words that were absent in the dictionary of the system of the morphological analysis appear as two morphological forms, e.g. *genoma* and *genomas* (*genome, genomes*). Note that usually they belong to the same cluster that is an additional proof of the correct functioning of the clustering algorithm.

The next step is evaluation of the obtained results. As usual in case of term detection and classification, evaluation is not trivial since there are no gold standards. This is an objective situation due to the fact that term extraction is really subjective.

In the paper [8], after manual evaluation of results, it is reported around 70% of precision.

We evaluate our results using the following calculations. Totally we obtained 270 terms, form these terms there are 31 verbs (underlined), i.e., there are 239 left. There are 19 errors that clearly are not terms of the domain (stroke out). Thus, we obtain precision of (239-19)/239=92.5%. If we consider terms of general science (48 terms) as errors, then we get the following precision (239-(19+48))/239 = 72%. Calculation of recall would be even more subjective; some domain dictionary might be used for this; still, the compilation of this dictionary keeps being subjective and does not guarantee its completeness.

4 Conclusions

In the paper we presented the method that allows for extraction of single-word terms. We consider that this can be the first step for multi-word term extraction. The proposed method is based on comparison with general reference corpus using log-likelihood similarity that is used for corpus comparison. In addition, we perform term clustering using *k-means* algorithm and cosine similarity measure.

We made experiments using texts of the domain of computer science. We analyzed in detail the obtained term list. Our results show precision of 92.5% using manual evaluation if we consider general scientific terms as correct ones and 72% if we consider them as errors.

As the main direction of future work we would like to mention:

- Comparison with various corpora for filtering terms that belong to domain of general science.
- Comparison of various log-likelihood measures.
- Extraction of multi-word terms using the extracted single-word terms as the candidates.

Acknowledgments. Work done under partial support of Mexican Government (CONACYT, SNI) and National Polytechnic Institute, Mexico (SIP, COFAA, PIFI), projects 20100668 and 20100772. We thank anonymous reviewers for their important comments.

References

1. Cimiano, P.: Ontology learning and population from text, algorithms, evaluation and applications. Springer, New York (2006)
2. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1993)

3. Gómez-Pérez, A., Fernandez-López, M., Corcho, O.: Ontological Engineering. Springer, London (2004)
4. Maedche, A., Staab, S.: Discovering conceptual relations from text. In: Proceedings of ECAI 2000 (2000)
5. Melchuk, I.A.: Lexical Functions in Lexicographic Description. In: Proceedings of VIII Annual Meeting of the Berkeley Linguistic Society, Berkeley, UCB, pp. 427–444 (1982)
6. Punuru, J.: Knowledge-based methods for automatic extraction of domain-specific ontologies. PhD thesis (2007)
7. Rayson, P., Berridge, D., Francis, B.: Extending the Cochran rule for the comparison of word frequencies between corpora. In: Purnelle, G., Fairon, C., Dister, A. (eds.) Le poids des mots: Proceedings of the 7th International Conference on Statistical analysis of textual data (JADT 2004), Louvain-la-Neuve, Belgium, March 10-12. Presses universitaires de Louvain, vol. II, pp.926–936. Presses universitaires de Louvain (2004)
8. He, T., Zhang, X., Xinghuo, Y.: An Approach to Automatically Constructing Domain Ontology. In: PACLIC 2006, Wuhan, China, November 1-3, pp. 150–157 (2006)
9. Uschold, M., Grunninger, M.: Ontologies: Principles, Methods and Applications. Knowledge Egineering Review (1996)