SciEsp: Structural Analysis of Abstracts Written in Spanish

Irvin Vargas-Campos, Fernando Alva-Manchego

Pontificia Universidad Católica del Perú,
Pattern Recognition and Applied Artificial Intelligence Research Group (GRPIAA),
Lima, Peru

{irvin.vargas, f.alva}@pucp.pe

Abstract. SciEsp is a tool for scientific writing in Spanish. Its objective is to help students when writing abstracts of scientific texts, such as a thesis or a dissertation. The tool identifies the different components of an abstract structure according to the guidelines of "good writing" proposed by the literature. Each sentence in the abstract is classified to one of six different rhetorical categories (background, gap, purpose, methodology, result, or conclusion), warning the writer of a possible missing component of the "optimal" structure. We manually annotated a corpus of abstracts from computer science theses and dissertations, and use it to train a Naive Bayes classifier that achieves an F1-measure of 0.65. We expect that SciEsp becomes a starting point for further projects in the area of supporting technologies for scientific writing in Spanish.

Keywords. Supporting technologies for scientific writing, argumentative zoning, supervised machine learning.

1 Introduction

Writing is not an easy task, even for people with a high level of formal education. It is common to read research papers, dissertations, theses, and other academic and professional texts with several writing errors [7]. To prevent this, students in their formation years should learn how to organize their ideas and prepare the information before writing an essay or complex scientific articles. A basic and fundamental technique to achieve this is writing an abstract [5]. To help in this matter, we present SciEsp, a software that helps students to write abstracts of scientific texts in Spanish, following a pre-defined structure. This software is based on SciPo [1], a tool for scientific writing in Portuguese, whose main goal is to support students when writing abstracts and introductions of academic texts.

In order to implement SciEsp, we first collected a corpus of abstracts of scientific texts in Spanish, and manually annotated their sentences as belonging to one of six categories: Context, Gap, Purpose, Methods, Results and Conclusion, following the recommendations given by Feltrim [2]. This data was then used to train and test a Naive Bayes classification model (AZEsp) whose goal is to automatically identify each category when presented with sentences of a new abstract. After several experiments and improvements, the classifier achieved a performance of 65.4 in F1-measure. We expect that this tool

The following sections describe SciPo, the tool on which SciEsp was inspired (2), how we collected and annotated the corpus of abstracts (3), the features extracted from the annotated data (4), the classification model AZEsp and its assessment (5), and some conclusions and future work (6).

2 Related Work: SciPo

SciPo [8] is a tool for scientific writing in Portuguese. Its purpose is to help students when writing abstracts and academic texts' presentations. The system supports text structuring according to the guidelines of "good writing" proposed by the literature. In addition, students can consult a database with real (and commented) examples of introductions, abstracts and dissertations of the Computer Science domain. SciPo evaluates

ISSN 1405-5546
doi: 10.13053/CyS-20-3-2463

abstracts and introductions. In each case, the system establishes a particular ideal structure for evaluating the text written by the user. An example of an abstract organization is presented in Table 1.

SciPo's automatic category recognition is performed by a statistical classifier similar to Teufel and Moens's Argumentative Zoning (AZ) [10], but ported to work on Portuguese abstracts [4]. AZ is rhetorical-level analysis of scientific articles which assigns a category (or "zone") to textual segments (sentence or group of sentences). Each category identifies the communicative function of the textual segment with respect to the whole paper.

SciPo-Farmácia [9], based on Scipo, is a tool for scientific writing in English. It is able to evaluate the organizational structure of abstracts, methods, results, discussions and conclusions of articles of different scientific domains. Just like SciPo, it also provides examples of each category and section of a scientific text, that students can use for further guidance.

3 Manual Annotation of Abstracts

For our corpus, we collected abstracts from theses and dissertations of different informatics areas, such as Information Systems, Information Technology, Software Engineering, Computer Science and Computer Engineering. For the analysis, we used 30 undergraduate theses from the Pontificia Universidad Católica del Perú, where most of them belonged to the Information Systems area; and 14 postgraduate theses from different foreign universities, such as the Autonomous University of Barcelona, the Polytechnic University of Catalunya, and others, where most of them belonged to the Computer Science area.

This decision was based on the fact that these scientific texts tend to have a well-defined organizational structure, and also because of its ease of collection. We then used the the annotation schema detailed in Table 1 (based in the structure presented by Feltrim [2]), and assigned a rhetorical category to each sentence in each abstract of the corpus.

The optimal structure would be made up of three main categories (*Purpose*, *Methodology* and *Result*) and three optional categories (*Background*, *Gap* and *Conclusion*). The category *Outline* should not be part of the abstract because it is indicative and not very informative; however, we considered this category in the schema because it is present in many scientific abstracts.

Table 1. Annotation schema

Background (B)
B1 Argue about the topic prominence
B2 Cite the results of previous research
B3 Present the importance of the research area
B4 Present the evolution over time of the research area
Gap (G)
G1 Cite problems/difficulties of a research area
G2 Cite the absence of previous research
G3 Cite negative aspects of other works
G4 Cite controversy between authors of the same research area
Purpose (P)
P1 Present the main objective
P2 Detail the main objective
P3 Describe the secondary objectives
Methodology (M)
M1 Describe methods and materials
M2 Justify the methodology used
M3 Indicate the criteria and conditions for the realization of the research
M4 Describe the dataset used
M5 Describe the procedure used for the evaluation and test of the results
Result (R)
R1 Describe the artifact (software, technique, etc)
R2 Present results of the experiments
R3 Present results of the evaluations
R4 Discuss about the results obtained
Conclusion (C)
C1 Describe conclusions
C2 Present contributions/value of research
C3 Present recommendations
Outline (O)
O1 Describe what will be presented in the article

After annotating the corpus, we noticed that 100% of the abstracts had the *Purpose* category. In the case of the *Background*, *Methodology* and *Results*, they were present in more than 50% of the texts, having an appearance percentage of 53.3%, 51.1% and 66.7% respectively. On other hand, the rest of the categories were present in less than

35% of the abstracts: *Gap* (33.3%), *Conclusion* (20%) and *Outline* (24%). In the case of Outline, the low percentage is a good parameter, because it is expected that abstracts do not contain this category.

4 Features Used for Rhetorical Category Identification

4.1 Overview of the Features

AZEsp is a classification model that will assign a possible rhetorical category, showed in Table 1, to each input sentence of a Spanish abstract. AZEsp receives sentences as vectors of features, making feature extraction is a crucial step in the application's pipeline. Table 2 shows a brief description of the set of features that AZEsp uses in order to classify the abstracts' sentences. These are based on the features presented by Feltrim [3] in her model AZPort for abstracts in Portuguese.

4.2 Detailed Description of the Features

We implemented a set of 6 features, based on the 8 features used by Feltrim [3].

4.2.1 Sentence length

It classifies a sentence as *short*, *medium* or *long* length, based on the number of words. The sentence is *short* if the number of words is less than 20, *long* if it is greater than 40 and *medium* if it is between these values. They were estimated based on the average sentence length present in our corpus.

4.2.2 Sentence location

It identifies the position occupied by a sentence in the abstract. We use five values for this feature:*first*, *second medium*, *penultimate* and *last*. These values represent common sentence locations for some specific categories of our scheme.

4.2.3 Presence of common expressions

It identifies the presence of a common expression in a sentence, and it classifies the sentence in a category based on the category of the expression contained. In order to make this possible, we recognized a set of about 100 common expressions in the corpus, and we manually classified them in the categories previously shown. Some examples of common expressions are presented in Table 3.

4.2.4 Verb tense, verb voice and presence of modal auxiliary

The features **Tense**,**Voice** and **Modal**, also called syntactic features, describe syntactic properties of the first finite verb of the sentence in the indicative or imperative mood. Because of the high probability of subjunctive verbs belonging to subordinate clauses, they are considered only when no other finite verb in the indicative or imperative mood is found. If no finite verb is found in the sentence, the three syntactic features take the value *noverb*. It is important to highlight that in determining the syntactic features, we considered both simple verbs (Example: "Los resultados muestran...") and phrasal verbs, also known as complex verbs, which include one or more auxiliary verbs to express the following:

1. Continuous aspect (*estar* + gerund), or Perfect aspect (*hacer* + participle). For example: "Este gran trabajo ha sido realizado para..."
2. Passive voice (*ser* + participle). For example: "ha sido realizado"
3. Modalization (*deber/poder/precisar/tener (que)/etc.* + infinitive)

The complex verb can also contain the pronoun *se* as a subject indeterminacy index or passive particle. In this paper, we will use the term "phrasal verb" to describe verbs in general, both simple and complex.

The **Tense** feature indicates the inflection of the verb (simple or complex) and it can take 14 values, including the value *noverb*. It uses *NOVERB* for verbal phrases, *IMP* for imperative sentences, or some identifier in the

Table 2. Summary of set of features

Feature	Description	Possible values
Length	What is the size of the sentence? (based on the limits of 20 to 40 words)	short, medium or long
Localization	What is the position of the sentence in the abstract?	first, second, medium, penultimate, last
Expression	Which rhetorical category is the common expression contained in the sentence?	B, G, P, M, R, C, S o noexpr
Tense	What is the tense of the first finite verb of the sentence?	IMP, PRES, PAST, FUT, COND, PRES-CPO, PAST-CPO, FUT-CPO, PRES-CT, PAST-CT, FUT-CT, PRES-CPO-CT, PAST-CPO-CT, FUT-CPO-CT or noverb
Voice	What is the voice of the first finite verb of the sentence?	passive, active or noverb
Modal	The first finite verb of the sentence is modal?	yes, no or noverb

Table 3. Examples of common expressions for each category

Category	Common Expression (Spanish)	Common Expression (English)
Background (G)	En el transcurrir de las últimas décadas... Actualmente Hoy en día En el ambiente de negocios de hoy En los ultimos años	In the passing of the last decades... Actually... Nowadays... In today's business environment... In recent years...
Gap (G)	Sin embargo... ...la problemática actual... No obstante...	However... ...the actual problem... Nevertheless...
Purpose (P)	El tema de tesis tiene como objetivo... El presente trabajo de tesis implementa... El presente trabajo de tesis presenta	The thesis aims... The present thesis implements... The present thesis presents...
Methodology (M)	Se emplearon metodologías... El análisis de software... El diseño de software... La implementación del software	The methodology used is... The software analysis... The software design... The software implementation...
Result (R)	La solución consiste en... A partir de los resultados... ...tiene las siguientes características...	The solution consists... From the results... ...has the following features...
Conclusion (C)	Se tiene como trabajo fututo... Este trabajo contribuye... Se concluye...	This project has as future work... This paper contributes... We conclude...
Outline (O)	La estructura de la tesis... Este documento ha sido estructurado... En el primer capítulo...	The structure of this thesis... This document has been structured... In the first chapter...

SimpleTense-(not)perfect-(not)continuous format, where *SimpleTense* indicates the tense of the finite component in the phrasal verb, *(not)perfect* indicates the presence of the auxiliary verb *haber* in the phrasal verb expressing the perfect aspect, and *(not)continuous* indicates the presence of auxiliary verb *estar* expressing the continuous aspect, see examples in Table 2.

The **Voice** feature indicates the verb voice, and it can take the values: *active*, *passive*, or *noverb*. The passive voice is understood in a broader sense, harboring certain forms and verbal constructions that are generally used to bypass an agent, i.e.:

1. Analytic passive voice (verb *ser* + participle),
2. Syntetic passive voice (it is done with the passive particle *se*),
3. Indeterminate subject indicated by the flexion of singular third person (it is done with the passive particle *se*).

The **Modal** feature indicates if there is a modal auxiliary in the phrasal verb and it can take the values: *yes*, *no* and *noverb*. The following verbs are considered as modal: *tener (que)*, *deber* and *poder*.

4.3 Extraction of the Features' Values

The features previously described are automatically extracted from the input text, through a process implemented in Java. As you can see in Figure 1, this process is divided in different stages: *Tokenization*, *Sentence delimitation*, *Expression identification*, *POS-Tagging* and *Syntactic processing*. Along the process, we used the Freeling library [6] in order to extract the features.

4.3.1 Tokenization

First, we divided the sentence into smaller independent units, i.e., words. From Freeling library, we used the *tokenizer* class, which receives plaintext and returns a list of *word* objects.

4.3.2 Sentence delimitation

Then, we grouped the obtained words in the first stage in order to generate the sentences of the text. From Freeling library, we used the *splitter* class, which receives the list of *word* objects obtained previously, and returns a list of *sentence* objects.

This second stage provides the required information to obtain features **Sentence length** and **Sentence location**. No problems were raised because of the presence of parentheses, brackets and braces, and abbreviation points (Example: Dr., Mr.) which can cause misinterpretations because they could be considered as endpoint.

4.3.3 Expression identification

To recognize the common expressions in text, we set up a group of common expressions divided in seven categories: *Background*, *Gap*, *Purpose*, *Methodology*, *Result*, *Conclusion*, *Outline*. Then, we programmed an algorithm that uses the set of expressions to find them in each sentence of the text. If the expression contained in the sentences belongs to the category C, the value of the feature **Expression** would be C.

4.3.4 POS-Tagging

This stage provides relevant information for the syntactic processing of the words of the text. From the Freeling library, we used the *maco* class, which receives a list of *sentence* objects and annotates morphologically each *word* object of each sentence given. It includes sub-modules such as detection of days, numbers, etc.

4.3.5 Syntactic processing

It was difficult to implement the feature extraction related to the verbs: **Tense**, **Voice** and **Modal**, because of the great morphological flexibility of the Spanish language and certain limitations of Freeling. We used the *tagger* class, which receives a list of *sentence* objects, and labels morphosyntactically and grammatically each *word* object of each sentence given.

The library helped to classify the words in adjectives, adverbs, determiners, nouns, verbs,

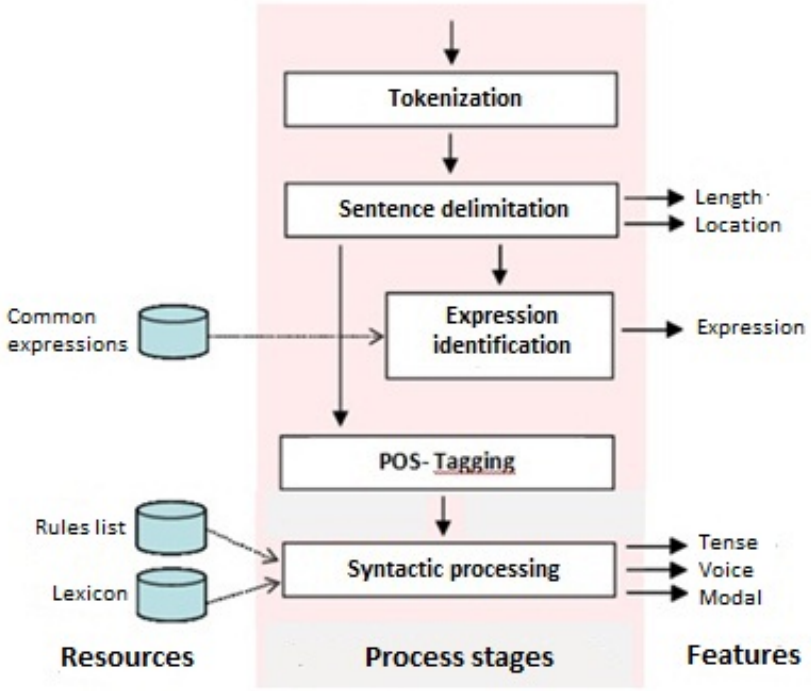

Fig. 1. Process stages of the extraction of features values by Feltrim [3]

pronouns, conjunctions, interjections, prepositions, etc. To obtain the features previously mentioned, we focused on the words classified as verbs. Unfortunately, the tool does not directly provide information that help to identify some tenses, some modals and passive voice.

For the **Tense** feature, we implemented an algorithm to identify phrasal verbs in present perfect tense, future, and others, mostly verbs composed of two or more words. For the **Modal** feature, we used a list of modals previously identified in the corpus in order to identify them. Finally, for the **Voice** feature, specifically Passive Voice, we detected the same problem: the identification of phrasal verbs; but for the Analytical Passive Voice, it was necessary to identify the verb "ser" accompanied by a participle verb.

5 AZEsp Classification Model

We used the Naive Bayesian classification model to estimate the probability that a sentence S belongs to a category C, based on the value of its features. The category C, that has a higher probability, is selected as output for sentence S. We performed three experiments to measure the classification capacity of the model. To execute these experiments, we used the software WEKA [1]. As mentioned previously, we used the Naive Bayes classifier, and the technique applied was *11-fold cross validation*, i.e., in each iteration, the classifier was trained using 40 abstracts and tested using 4 abstracts.

5.1 Automatic Annotation Results

In order to measure the performance of the classifier model, we took as reference the *F-measure*, defined as

$$\frac{2 \times P \times R}{P + R},$$

[1] Weka is a collection of machine learning algorithms for data mining tasks [11]

where P means *Precision*[2] and R means *Recall*[3]. Both parameters are provided by WEKA. We also considered the number of *Correctly Classified Instances.*.

In the first experiment, we considered all defined categories and all the features, except for *Expression*. In this setting, AZEsp has a good performance classifying categories as *Methodology* (F-Measure=0.522) and *Background* (F-Measure=0.474), but no *Gap* (F-Measure=0) and *Conclusion* (F-Measure=0.095). This is because the number of sentences manually categorized as *Gap* or *Conclusion* is very low. Overall, AZEsp classified correctly 40.26% of the sentences.

In the second experiment, we used the first experiment setup, but we didn't consider the category *Outline*, because it should not be part of the abstracts of scientific texts. AZEsp had slightly better results, classifying correctly 44.26% of the sentences.

In the third experiment, which had the best results, we considered all categories, and we included the feature *Expression*, but we did not consider the feature *Modal* because it worsened a little the model performance. As we can see in Table 4, the number of successful categories classified improved. Because of the inclusion of the feature *Expression*, the classifier model did a clearer distinction between the different categories.

Table 4. Confusion Matrix: human vs automatic annotation

		Machine						
	Cat.	B	G	P	M	R	C	O
Human	B	**40**	2	0	18	4	0	0
	G	4	**22**	1	9	4	0	0
	P	4	3	**39**	11	4	0	0
	M	1	4	0	**94**	17	1	0
	R	3	1	1	35	**38**	0	0
	C	0	1	2	7	4	**4**	0
	O	0	0	0	14	3	0	**62**

[2]Number of sentences correctly classified as C out of number of sentences that the model classified as C.

[3]Number of sentences correctly classified as C out of total number of sentences categorized as C.

Additionally, Table 5 shows that the F-measure for all categories increased significantly, even the one for the category *Gap* increased to 0.603. In this last experiment, AZEsp classified correctly 65.4% of the sentences. These results show the relevance of the feature *Expression* for identifying the rhetorical categories above all the other features.

Table 5. Performance evaluation per category

Category	Precision	Recall	F-Measure
Background	0.769	0.625	0.69
Gap	0.667	0.55	0.603
Purpose	0.907	0.639	0.75
Methodology	0.5	0.803	0.616
Result	0.514	0.487	0.5
Conclusion	0.8	0.222	0.348
Outline	1	0.785	0.879

Finally, in the fourth experiment, we used the third experiment setup, but we used SMO classification. AZEsp had slightly worse results, classifying correctly 62.58% of the sentences.

6 Conclusions and Future Work

In this paper, we introduced SciEsp: a tool to help students write abstracts of scientific texts. We collected and annotated a corpus of computer science abstracts, and use it to build a classifier (AZEsp) to automatically identify the rhetorical structure of a given abstract in Spanish. In its current state, AZEsp has an accuracy of 65%, which is well-enough for the SciEsp environment. However, the classifier's performance could be improved. One way to do it would be by generalizing the list of common expressions, using regular expressions. Additionally, we could implement the feature *History* used in SciPo, which indicates the category of the sentence immediately before to the one being analyzed. We expect that the work presented here constitutes the starting point for other projects in the same field with a wider scope. For example, other projects could cover the remaining sections of a scientific article, such as Introduction. Also, they could analyze

other discourse aspects of the text, such as cohesion or coherence.

References

1. **Antiqueira, L., Feltrim, V., & Nunes, M. (2003).** Projeto e implementação do sistema SciPo. Technical report, Relatórios Técnicos do ICMC-USP.
2. **Feltrim, V., Aluísio, S., & Nunes, M. (2003).** Analysis of the rhetorical structure of computer science abstracts in Portuguese. *Proceedings of the Corpus Linguistics*, volume 16, pp. 212–218.
3. **Feltrim, V., Nunes, M., & S.M., A. (2001).** *Um Corpus de Textos Científicos em Português para a Análise da Estrutura Esquemática*. Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional NILC - ICMC-USP.
4. **Feltrim, V., Pelizzoni, J., Teufel, S., & Nunes, S., M.G.V. add Aluísio (2004).** Applying argumentative zoning in an automatic critiquer of academic writing. *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA)*, Springer, pp. 214–223.
5. **Huaman, M. (2005).** Cómo escribir un artículo científico. *Filología, Lingüística y Literatura*, Vol. XXXI, No. 1, pp. 267–295.
6. **Padró, L. & Stanilovsky, E. (2012).** FreeLing 3.0: Towards wider multilinguality. *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, ELRA, Istanbul, Turkey.
7. **Sanchez, C. (2005).** Los problemas de redacción de los estudiantes costarricenses: Una propuesta de revisión desde la lingüística del texto. *Revista de Filología y Lingüística de la Universidad de Costa Rica*, Vol. 31, No. 5, pp. 267–295.
8. **SciPo (2013).** SciPo. Retrieved April 20, 2013 from http://www.nilc.icmc.usp.br/scipo.
9. **Scipo-Farmacia (2013).** Scipo-Farmacia. Retrieved April 20, 2013 from http://www.nilc.icmc.usp.br/scipo-farmacia.
10. **Teufel, S. & Moens, M. (2002).** Summarising scientific articles — experiments with relevance and rhetorical status. *Computational Linguistics*, Vol. 28, pp. 409–446.
11. **WEKA (2013).** Weka 3. Retrieved November 09, 2013 from http://www.cs.waikato.ac.nz/ml/weka/.

Irvin Vargas-Campos has a BSc in Computer Engineering from the Pontifical Catholic University of Peru (PUCP), where he also works as a teacher's assistant and analyst at the Information Technology Department (DTI). His research interests involve using Natural Language Processing and Machine Learning techniques to develop applications that help students write different types of texts properly.

Fernando Alva-Manchego has a Masters degree in Computer Science from the University of Sao Paulo, is a professor at the Pontifical Catholic University of Peru (PUCP) and a member of the Pattern Recognition and Applied Artificial Intelligence Group (GRPIAA). His research interests involve using Natural Language Processing and Machine Learning techniques to develop applications that contribute to the teaching - learning process, as well as to allow information access to people with low literacy levels.

Article received on 14/01/2016; accepted on 02/03/2016.
Corresponding author is Irvin Vargas-Campos.