Chapter 18

Argumentative Zoning Applied to Critiquing Novices' Scientific Abstracts

Valéria D. Feltrim[1], Simone Teufel[2], Maria das Graças V. Nunes[1] and Sandra M. Aluísio[1]
[1]University of São Paulo
NILC,ICMC, Universidade de São Paulo,
Av. do Trabalhador São-Carlense, 400
13560-970, São Carlos, SP, Brasil.
Email: {vfeltrim, gracan, sandra}@icmc.usp.br

[2]University of Cambridge
Computer Laboratory, University of Cambridge,
JJ Thomson Avenue, Cambridge CB3 0FD, U.K.
Email: Simone.Teufel@cam.ac.uk

Abstract

We present a system that applies Argumentative Zoning (AZ) (Teufel and Moens, 2002), a method of determining argumentative structure in texts, to the task of advising novice graduate writers on their writing. For this task, it is important to automatically determine the rhetorical/argumentative status of a given sentence in the text. On the basis of this information, users can be advised that a different sentence order might be more advantageous or that certain argumentative moves are missing. In implementing such a system, we had to port AZ from English to Portuguese, as our system is designed to help the writing of Brazilian PhD theses in Computer Science. In this chapter, we report on the overall system, named SciPo, the porting exercise, including a human annotation experiment to verify the reproducibility of our annotation scheme, and the intrinsic and extrinsic evaluation of the AZ module of the system.

Keywords: academic writing, Argumentative Zoning, machine learning.

1. Introduction

It is widely acknowledged that academic writing is a complex task, since it involves the complexities of the writing process as well as those specific to the academic genre (Sharples and Pemberton, 1992). It can be even harder for novice writers, who are usually not well acquainted, if at all, with the requirements of the academic genre. Even when the basic guidelines on scientific writing are explicit and known, it can be very difficult to apply them to a real text. To improve the quality of academic texts produced by novice and/or non-native writers, a number of writing tools have been described in the literature (Sharples et al, 1994; Broady and Shurville, 2000; Narita, 2000; Aluísio et al., 2001).

The project SciPo (short for Scientific Portuguese) aims at analysing the rhetoric structure of Portuguese academic texts — in terms of schematic structure, rhetorical strategies and lexical patterns — to derive models for supporting the creation and evaluation of computational writing tools and its outcomes. This project is currently being developed at *Núcleo Interinstitucional de Lingüística Computacional* (NILC)[1], University of São Paulo. To make it feasible, the analysis has focused on specific sections of theses in Computer Science, namely the abstract and the introduction, which are the most studied in the literature (Swales, 1990; Weissberg and Buker, 1990; Liddy, 1991; Santos, 1996). In conjunction with conclusions, these particular sections have also been pointed out as the most difficult ones to be written in a questionnaire applied to graduated students of Computer Science from University of São Paulo. The reasons for working on this kind of text and domain were threefold: firstly, in the Brazilian University system, theses have to be written in Portuguese, unlike research articles, which are preferably written in English; secondly, there exists a high degree of standardization in Computer Science texts, as in other scientific research areas; and thirdly, SciPo's developers are familiar with the Computer Science domain, as it is being developed in a Computer Science department.

As the approach being followed is corpus-based, an analysis of a specific corpus was carried out by human annotators, based mainly on Swales's (1990) and Weissberg and Buker's (1990) models. The used annotation scheme has the following rhetorical categories: `Background`, `Gap`, `Purpose`, `Methodology`, `Results`, `Conclusion` and `Outline`. Examples of sentences for each category are presented in Figure 1. For convenience, the examples are presented in English although our corpus is in Portuguese and were collected from Anthony and Lashkia (2003) (except the example sentence for Outline). The results of this analysis have been used as basis for a computational model using (good and bad) examples and rules. Moreover, this analysis helped us to understand the problems novice writers face when writing in a new genre. We have identified some writing problems that are specific to the academic genre, such as misuse of lexical patterns and verbal tenses, inefficient organization and inappropriate emphasis on some specific components. On the basis of these results, we believe that especially novice writers may benefit from a writing support tool that provides critiques about text structure, a repository of good and bad examples of structure, writing strategies and lexical patterns. In the next section we introduce the SciPo system, focusing on its architecture and linguistics resources.

2. The SciPo System

Inspired by the Amadeus system (Aluísio et al., 2001), SciPo is a system whose ultimate goal is to support novice writers in producing academic writing in Portuguese, specially abstracts and

[1] http://www.nilc.icmc.usp.br/nilc/index.html

introductions of Computer Science theses. Its current main functionalities can be summarized as: (a) browsing and searching on a base of authentic thesis abstracts manually annotated according to our structure model (Feltrim et al., 2003) for all occurrences of a specific rhetorical strategy and/or structural component; (b) browsing and searching on a base of authentic thesis introductions manually annotated according to an adaptation of Aluísio and Oliveira Jr.'s (1996) model, in the same way of the abstracts base; (c) support to build a structure for the writer to use as a starting point for his/her text; (d) application of critiquing rules to the created structure; and (e) recovery of authentic cases that are similar to the writer's structure. Also, the existing lexical patterns from the case base are highlighted allowing the writer to easily add such patterns to a previously built structure. Examples of lexical patterns are underlined in Figure 1.

1 **Background**
"The research article (RA) or paper is one of the most important genres that both scientists and engineers will write."

2 **Gap**
"When faced with the tasks of reading and writing a complex technical paper, many nonnative scientists and engineers (...) lack an adequate knowledge of commonly used structural patterns at the discourse level."

3 **Purpose**
"In this paper, we propose a novel computer software tool that can assist these people in the understanding and construction of technical papers (...)."

4 **Methodology**
"The software uses a supervised learning approach, in which the system first "learns" the characteristic features of text structure in a particular discipline using a small number of training examples."

5 **Results**
"We can see that the system performs consistently across the different data sets, with an average accuracy of 68%."

6 **Conclusion**
"The system is tested using research article abstracts and is shown to be fast, accurate, and useful aid in the reading and writing process."

7 **Outline**
"In the next section we present the contextualization of this work and details about the used methodology."

Figure 1. Example sentences for each category with lexical patterns underlined.

SciPo contains four knowledge bases, namely the Abstracts Case Base, Introductions Case Base, Rules and Similarity Measures, and Critiquing Rules. As explained before, the Case Bases were built through manual annotation, based on predefined rhetorical schemes. The Abstract Case Base has 52 instances of schematic structures of authentic abstracts, describing the rhetorical components, strategies and lexical patterns of each case. The Introduction Case Base has 48 instances and represents the same kind of information described for abstracts. The Rules and Measures of Similarity are based on similarity rules among lists (pattern matching) and on nearest neighbours matching measure (Kriegsman and Barletta, 1993). These rules are used in the case recovery process, when a search is performed according to the writer's request of a specific schematic structure. The Critiquing Rules are based on prescriptive guidelines for good writing in the literature and on structural problems observed in the annotated corpus, as an attempt to anticipate and correct problematic structural patterns the writer might construct. The rules cover two distinct types of problems: content deviations (absence of components) and order deviations (occurrence order of components and strategies inside the overall structure). Thus, we have four classes of rules: content critiques, order critiques, content suggestions and order suggestions. We use critiques for serious problems such as, detecting absence of the purpose component, or for generating suggestions for structures that do not have serious problems but can be enriched by adding new components and/or reorganizing existing ones.

An example of an abstract with poor structure is [P M B G P], where the main purpose (first P) is followed by the Methodology (M) used to accomplish that purpose. Next, the most natural move would be to present results; however, the writer used a background component, followed by a gap (B G), providing more detail of the previously stated purpose and the introduction of yet other purposes. The presence of background and gap in the middle of the abstract, separating the main purpose from its subsequent detail, confuses the reader, who may lose track of the main purpose of the related research. Also, the sequence [M B] disrupts the cohesion of the text and may cause the reader to feel that "something is missing".

Using the resources mentioned above, the writer can build his/her own structure by choosing components/strategies from a predefined list, get feedback from the system until an acceptable structure has been built, recover authentic similar examples and use example lexical patterns (as found in the corpus) in his/her own writing. These can be very helpful for the writer to organize the structure of his/her text before the actual writing, but once the text has been constructed, the system cannot say anything about its structure. To overcome this drawback, we decided to provide a critiquing tool capable of giving feedback on the organization of the text after its writing, instead of just aiding its composition. For a tool to supply the writer with such information, it has to be able to elicit the schematic structure of texts automatically. Such analysis has been proved to be feasible by means of a text classifier (Teufel and Moens, 2002; Burstein et al., 2003; Anthony and Lashkia, 2003). With information about the rhetorical status of each textual part, SciPo could apply the critiquing rules previously mentioned to actual texts, instead of building structures. Figure 2 presents a simplified version of the SciPo's architecture, including the aforementioned critiquing tool.

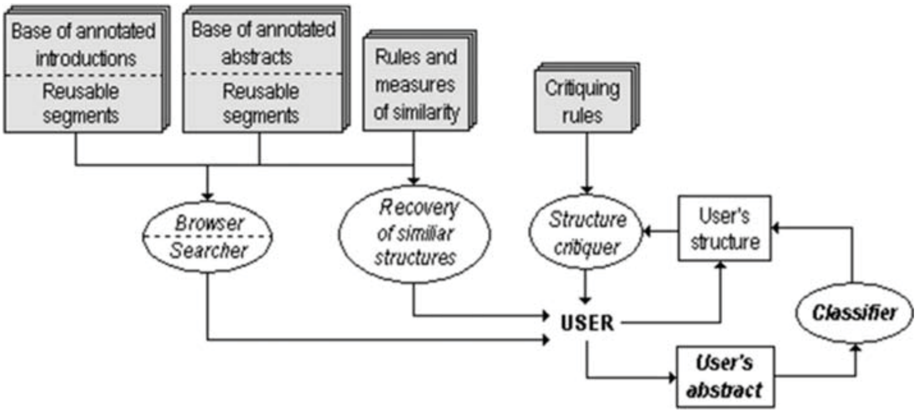


Figure 2. Simplified version of SciPo's architecture.

After analyzing previous work reporting on this kind of classification task (Teufel and Moens, 2002; Burstein et al., 2003; Anthony and Lashkia, 2003), we have found that Teufel and Moens's approach, named Argumentative Zoning, might suit our purpose better, considering the category scheme we wanted to use and the cost of adapting the feature extraction process to work on

Portuguese texts. To evaluate this assumption, we have run an experiment using a feature extraction pipeline similar to AZ’s and the Weka implementation of a Naïve Bayes classifier (Witten and Frank, 2000). As we have got encouraging results, we decided to implement our own classifier, named AZPort (as it is based on AZ), so we could take the context into account during classification. Details on both experiments are presented in the next section. We also present information about the manual annotation of the abstracts used as training material.

3. Argumentative Zoning for Portuguese Texts

Argumentative zoning is the task of breaking a text containing a scientific argument into linear zones (i.e. contiguous sentences) of the same argumentative status, or zones of the same intellectual attribution (Teufel and Moens, 2000). The zone segmentation is done automatically by a statistical classifier, based on textual features that can be readily read off the text. The parameters of the statistical model, a simple Naive Bayesian classifier, are learned from human-annotated texts. For Portuguese abstracts, we followed a similar approach, adapting the textual features and the set of target categories to fit our purposes. We assumed the seven categories presented in Figure 1 as the target ones.

The first step was to select the set of features to be applied in the experiment. Considering that we want to classify abstracts sentences, we decided to use seven features, derived from the original AZ feature set, namely:

- Sentence length (*Length*);
- Sentence location (*Location*);
- Presence of citations (*Citation*);
- Presence of formulaic expressions (*Formulaic*);
- Verb tense (*Tense*);
- Voice (*Voice*);
- Presence of modal verb (*Modal*).

The *Length* feature classifies a sentence as short, medium or long length, based on two thresholds (20 and 40 words) that were estimated using the average sentence length found in our corpus.

The *Location* feature identifies the position occupied by a sentence within the abstract. We use four values for this feature: first, medium, 2ndlast and last. We believe that these values characterize common locations for some specific categories of our scheme. In fact, experiments using other values showed these to be the best ones.

The *Citation* feature flags the presence of citations in a sentence. As we are not working with full texts, it is not possible to parse the reference list in order to identify self-citations. Nevertheless, as we are dealing with a theses corpus, that usually may not contain self-citations, we believe that such distinction would not affect the classification task.

The *Formulaic* feature identifies the presence of a formulaic expression in a sentence and the category (within our category scheme) to which an expression belongs. In order to recognize these expressions, we built a set of 377 regular expressions. The sources for the construction of the regular expressions set came from corpus observations, and the literature (translated into Portuguese).

Due to the relatively abundant inflectional morphology of Portuguese, much of the porting effort went into adapting verb-syntactic features. The *Tense*, *Voice* and *Modal* features report syntactic properties of the first finite verb phrase in indicative or imperative mood. Only if no indicative or imperative verb form is found, a subjunctive form is considered. This decision was made to avoid focusing on a verb of a subordinate clause. *Tense* may assume 14 values, namely 'noverb' for verbless sentences, 'Imp' for imperatives, or some identifier in the format 'SimpleTense-(not)perfect-(not)continuous', where 'SimpleTense' refers to the tense of the finite component in the verb phrase, and '(not)perfect/(not)continuous' flags the presence of perfect/continuous auxiliary "*ter* | *haver* / *estar*". As verb inflection in Portuguese has a wide range of simple tenses – many of which are rather rare in general and even absent in our corpus – we collapsed some of them. As a result, 'SimpleTense' may assume one single value 'Past'/'Future', to the detriment of the three/two morphological past/future tenses. In addition, 'SimpleTense' neutralizes mood distinction. The *Voice* feature may assume 'noverb', 'Passive' or 'Active'. Passive voice is understood here in a broader sense, collapsing some Portuguese verb forms and constructs that are usually used to omit an agent, namely (i) regular passive voice (analogous to English, by means of auxiliary "*ser*" plus past participle), (ii) synthetic passive voice (by means of passivizating particle "*se*") and (iii) a special form of indeterminate subject (also by means of particle "*se*"). The *Modal* feature may assume 'noverb' or flag the presence of a modal auxiliary.

The next step was to determine the corpus that would be used as training material, as AZ uses supervised learning. We decided to use a corpus of 52 abstracts in Portuguese (366 sentences) from Computer Science theses collected in a previous study (Feltrim et al., 2003). In order to verify the reproducibility of our annotation scheme (Figure 1) and whether the annotated corpus could be used as valid training material, we performed an annotation experiment.

Based on our annotation scheme and using specific annotation guidelines similar to the original AZ guidelines, we trained three human annotators, one of them being the first author. They were already knowledgeable of the corpus domain and familiar with scientific writing, so the training focused on defining each category and interpreting the stated guidelines. Our corpus presents a high number of sentences with "overlapping rhetorical roles", which often leads to doubt about the correct category to be assigned. Therefore, the full understanding of the guidelines is very important since they state strategies to deal with conflicts between categories. We used 6 abstracts in the training phase, which was performed in three rounds, each round consisting of explanation, annotation, and discussion. We found that the training phase was crucial to calibrate the annotators' knowledge about the annotation task. After training, the annotators were asked to annotate 46 abstracts sentence by sentence, assigning exactly one category per sentence. We used the *Kappa* coefficient *K* (Siegel and Castellan, 1988) to measure reproducibility among *k* annotators on *N* items. In our experiment, items are sentences. The use of the *Kappa* measure is appropriated in this kind of task since it discards random agreement. The formula for the computation of Kappa is:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is pairwise agreement (the proportion of times judges agree) and *P(E)* is random agreement, (the proportion of times that we would expect judges to agree by chance). *Kappa* varies between -1 and 1. It is -1 for maximal disagreement, 0 for if agreement is only as would be expected by chance annotation following the same distribution as the observed distribution, and 1 for perfect agreement.

The results show that our scheme is reproducible (*K*=0.69, *N*=320, *k*=3). Considering the subjectivity of this task, these results are acceptable. In a similar experiment, (Teufel et al., 1999) measured the reproducibility of their scheme as slightly higher (*K*=0.71, *N*=4261, *k*=3). One reason why our agreement rate is lower than theirs might be that our scheme refines their Own category into more specific categories Methodology, Results and Conclusion, increasing the complexity of the task. Collapsing these three categories increases our agreement significantly (*K*=0.82, *N*=320, *k*=3). When comparing our results to (Teufel et al., 1999), it is important to bear in mind that our corpus is much smaller. Based on these results, we concluded that trained humans can distinguish our set of categories and thus the data resulting from this experiment are reliable enough to be used as training material for an automatic classifier.

3.1 Experiment Using Weka

For the extraction of features values, the abstracts were automatically pre-processed, starting with the segmentation into sentences using XML tags and tokenization. Citations in running text were also marked with an XML tag and the sentences were POS-tagged according to a simplification of the NILC tagset (Aires et al., 2000). The target categories are the seven categories in the annotation experiment described above. As baselines, we considered a random choice of categories weighted by their distribution in the corpus (Baseline 1) and classification as the most frequent category (Baseline 2). The categories distribution in our corpus is presented in Figure 3.

Category	%
Background	21
Gap	10
Purpose	18
Methodology	12
Result	32
Conclusion	5
Outline	2

Figure 3. Distribution of components in the abstract corpus.

We used the Naive Bayesian classifier from the Weka system (Witten and Frank, 2000) for our experiments and the performance was measured by comparing the system's prediction with one human annotation. We assumed the annotation performed by one of the subjects in the previous annotation experiment as our "gold standard" and used it as training material. The agreement between the system and the human annotator was *K*=0.58, when compiled with a 13-fold cross-validation, and *K*=0.56 when using 66% of the data for training and the remainder for testing. This is encouragingly high amount of agreement (compared to Teufel and Moens' figure of *K*=0.45). Such a good result might be in part due to the fact that we are dealing with abstracts instead of longer texts (full papers). This result is also better than the baselines (Baseline 1: *K*=0 and Baseline 2: *K*=0.26).

Further analysis of our results shows that, apart from category Outline, the classifier performs well on the others categories, cf. the confusion matrix in Figure 4. This result is not surprising, since we are dealing with a corpus of abstracts, which is low in outline sentences (total of 6 sentences in the whole corpus). Many machine learning algorithms, including the Naïve Bayes classifier, perform badly on infrequent categories due to the lack of sufficient training material. Regarding the other categories, the best performance of the classifier is for Purpose (*F-measure*=0.82), followed by Gap (*F-measure*=0.70). We calculated *F-measure* as:

$$\frac{2*P*R}{P+R}$$

where *P* is *precision* and *R* is *recall*. We attribute the high performance on these categories to the presence of strong discourse markers on these kinds of sentences (modelled by feature *Formulaic*).

We were also interested in measuring the impact of taking the context of the sentence into account in our classification task. It is known that some argumentative zones tend to follow other particular zones. This property is even more apparent in self-contained texts such as abstracts (Feltrim et al., 2003). In our corpus, some particular sequences of argumentative zones are very frequent. For example, the pattern `Background` followed by `Gap`, with repetition or not, and then followed by `Purpose`, i.e. `((BG)|(GB)+)P`, occurs in 30.7% of the corpus. So, we decided to use a context feature *History* that holds the category of the sentence classified previously, as was used in the original AZ.

		Weka Naive Bayesian Classifier							
		`B`	`G`	`P`	`M`	`R`	`C`	`O`	Total
Human	`B`	**48**	7	0	2	19	0	1	77
	`G`	7	**24**	0	0	5	0	0	36
	`P`	3	0	**52**	0	9	1	0	65
	`M`	1	0	0	**27**	17	0	0	45
	`R`	6	1	9	4	**93**	3	1	117
	`C`	0	0	0	0	14	**6**	0	20
	`O`	0	0	0	0	5	1	**0**	6
	Total	65	32	61	33	162	11	2	366

Figure 4. Confusion matrix: Weka Naive Bayesian classifier vs. human.

During training, the value of *History* can be calculated by simple corpus observation. For unseen texts, however, it has to be estimated as a second pass process during testing, using the posterior probabilities of all categories (values) obtained for the previous sentence. As Weka does not give access to the posteriors, and as this would facilitate the integration of the classifier into the SciPo system, we implemented our own Naïve Bayes classifier, named AZPort. For the estimation of the feature *History*, we performed a beam search with width three among the candidate categories for the previous sentence to reach the most likely classification, following Teufel and Moens (2002). In the next section, we present the classification results of AZPort.

3.2 Experiment Using AZPort

AZPort is a Naïve Bayesian classifier that renders each input sentence a set of possible rhetorical status with their respective estimated probabilities. Similarly to the AZ classifier, it estimates the probability *P* that a sentence *S* has category *C* given the values of its feature vector *V*. The category with the highest probability is chosen as the output for the sentence. The implemented classifier is presented in Figure 5.

Again, the results of classification were compiled by applying 13-fold cross-validation to our 52 abstracts (training sets of 48 texts and testing sets of 4 texts). We considered the same baselines of the previous experiment. Comparing our classifier, trained with the full feature pool (the seven features described above plus *History*), to one human annotator, the agreement reaches *K*=0.65 (system accuracy of 74%). This is a better result than the previous one (*K*=0.58) and also much better than Baseline 1 (*K*=0 and accuracy of 20%) and Baseline 2 (*K*=0.26 and accuracy of 32%). It shows that taking the context into account is a helpful heuristic, as it improved our result significantly, by 12%.

$$P(C \mid F_0,\ldots,F_{n-1}) \approx P(C)\frac{\prod_{j=0}^{n-1} P(F_j \mid C)}{\prod_{j=0}^{n-1} P(F_j)}$$

$P(C \mid F_0,\ldots,F_{n-1})$: Probability that a sentence has target category *C*, given its feature values $F_0, \ldots, F_{n-1}$;

$P(C)$: (Overall) probability of category *C*;

$P(F_j \mid C)$: Probability of feature-value pair F_j, given that the sentence is of target category *C*;

$P(F_j)$: Probability of feature value F_j;

Figure 5. Naïve Bayesian classifier (Teufel and Moens, 2002).

Looking at the contribution of single features, the power of the feature *History* can be confirmed. In Figure 6, the second column gives the predictiveness of the feature on its own, in terms of *Kappa* between the classifier (actually, the 13 classifiers obtained in cross-validation) and one human annotator (gold standard). As can be observed, *Formulaic* is still the strongest feature, followed by *History*. Apart from these two, all other features are outperformed by both baselines. Syntactic features -- *Tense*, *Voice* and *Modal* -- and *Citation* are the weakest. We believe that the *Citation* feature would perform better in other kind of text than abstracts (e.g. introductions). The third column in Figure 6 gives *Kappa* coefficients for experiments using all features except the one given in the first column. As shown, apart from syntactic features, all features contribute some predictiveness in combination with others.

Feature	Alone	Left out
Length	-0.106	0.620
Location	-0.047	0.624
Citation	-0.272	0.630
Formulaic	0.557	0.345
Tense	-0.166	0.642
Voice	-0.018	0.644
Modality	-0.287	0.650
History	0.251	0.540
Baseline 1 (Random by distribution): *K*=0		
Baseline 2 (most frequent category): *K*=0.26		

Figure 6. Potential of individual features in 13-fold cross-validation.

		AZPort							
		B	G	P	M	R	C	O	Total
Human	B	**57**	10	2	1	7	0	0	77
	G	11	**23**	0	0	2	0	0	36
	P	6	1	**49**	0	8	1	0	65
	M	5	0	0	**26**	14	0	0	45
	R	2	2	0	9	**101**	3	0	117
	C	0	0	0	0	9	**10**	1	20
	O	0	0	0	0	5	1	**0**	6
	Total	81	36	51	36	146	15	1	366

*Figure 7. Confusion matrix: **AZPort** automatic annotation in 13-fold cross-validation vs. human.*

The analysis of the confusion matrix presented in Figure 7 shows that AZPort and Weka Naive Bayes has a similar behaviour. However, AZPort is more accurate. Its best performance is for Purpose sentences (*F-measure*=0.84), followed by Result sentences (*F-measure*=0.77). The worst performance is for Outline (*F-measure*=0). As pointed out earlier, the classifier performed badly on this category due to the lack of sufficient training material. Figure 8 presents *precision*, *recall* and *F-measure* for each category.

The results for the AZPort classifier are reasonably in agreement with our previous experimental results for human classification. We also observed that the confusion categories of the automatic classification are similar to the confusion categories of our human annotators. As can be observed in Figure 7, the classifier has problems in distinguishing the categories Methodology, Result and Conclusion and so do our human annotators. As mentioned previously, collapsing these three categories in one raises the human agreement considerably, which suggests distinction problems amongst these categories even for humans.

Category	***Precision***	***Recall***	***F-Measure***
Background	0.70	0.74	0.72
Gap	0.64	0.64	0.64
Purpose	0.96	0.75	0.84
Methodology	0.72	0.58	0.64
Result	0.69	0.86	0.77
Conclusion	0.67	0.50	0.57
Outline	0	0	0

Figure 8. Precision, Recall and F-Measure per category.

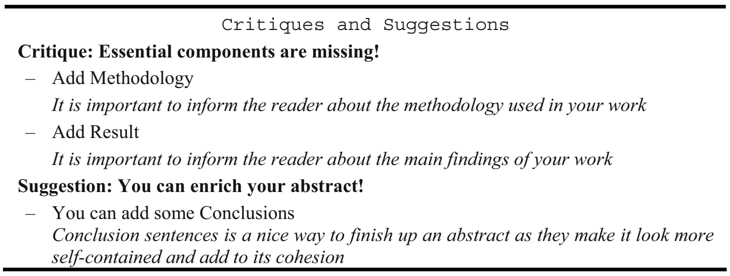
We concluded that the performance of AZPort, although lower than human, is promising and acceptable to be used as part of SciPo's critiquing tool. In the next section, we describe briefly the critiquing tool and how it works on unseen abstracts. We also report on an evaluation experiment.

4. Evaluation of SciPo's Critiquing Tool

One of the main ideas underlying SciPo's critiquing tool is that a good abstract must provide factual and specific information about a work. Thus, our aim is to help academic writers to produce more "informative" abstracts, in which the reader is likely to learn quickly what is most characteristic of and novel about the work at hand.

As previously mentioned, the critiquing tool is composed of two agents: a classifier, which detects the schematic structure elements of an abstract; and a critiquing component that analyzes the detected structure. We use AZPort for the classification task and the critiquing rules described in Section 2 for the critiquing component. Figure 9 presents the critiques and suggestions generated by the critiquing tool when analysing an abstract with the structure [B G P].

In order to evaluate how well real users would interact with the critiquing tool and to which extent it would improve their writing, we made an experiment with four students who had just finished their Master's dissertation in Computer Science at the University of São Paulo. We were also interested in observing the impact of the mistakes made by the classifier on the overall result of the critiquing tool.



Critiques and Suggestions

Critique: Essential components are missing!

- Add Methodology
  It is important to inform the reader about the methodology used in your work
- Add Result
  It is important to inform the reader about the main findings of your work

Suggestion: You can enrich your abstract!

- You can add some Conclusions
  Conclusion sentences is a nice way to finish up an abstract as they make it look more self-contained and add to its cohesion

Figure 9. Output of the critiquing tool when analyzing the structure [B G P].

The students were asked to use SciPo to rewrite the abstracts of their dissertations. One of the developers/authors was present, but intervened only when prompted by the student. Before starting the experiment, all students were asked to read a document explaining SciPo's main funcionalities, as we were not interested in assessing system interface, but rather, its effectiveness as a critiquing tool. After the evaluation, all four students filled in a questionnaire, which asked for general impressions about the system.

During the evaluation experiment, the students were asked to input their abstracts into the classifier for structure detection. Before submitting the detected structure to the critiquer, the students could correct the automatic classification, if desired. Two students made corrections, while the other two assumed the classification as totally correct. The four students got suggestions/critiques from the system and changed their abstract to some extent.

Although aware of the accuracy rate of the classifier, the students considered its results very reliable and this affected the way they interacted with the tool. The classifier made mistakes in three of the four abstracts, with different impact on the resulting critiques. The observerd misclassifications were: (a) Gap vs. Background, (b) Purpose vs. Background, (c) Methodology vs. Result and (d) Purpose vs. Conclusion. (a) and (c) occurred simultaneously in a single abstract (Student 1). (b) and (d) occurred in two different abstracts (Students 2 and 3). Student 1 did not correct any of the two mistakes. Students 2 and 3 did correct the classification mistakes on their abstracts. We believe that this difference in behaviour might be caused by the kind of misclassifications made by the system. In (b) and (d), the classifier confounded very dissimilar categories and thus less problematic to be corrected, as the writer is likely to perceive such mistakes. On the other hand, mistakes like (a) and (c) are a major problem, as these categories are hard to distinguish even for trained annotators. Thus, Student 1 was not able to perceive the mistakes and then accepted the automatic results as correct. This caused the system to emit unhelpful critiques and suggestions on Student 1's abstract.

Regarding the questionnaire, the four students reported their experience with SciPo as positive. As commented above, they considered the classifier reliable. They also considered the critiques and suggestions relevant, except for one student that considered the suggestions not relevant. All students evaluated SciPo as a useful tool and reported their intention of using it again on a real situation.

To evaluate if there were improvements in the writing, we used two sets of abstracts: the original ones and the ones rewritten using SciPo. Then we asked an expert judge, experienced at academic writing, to analyse both sets and point out if there were any improvements in the rewritten abstracts regarding structure. The expert was knowledgeable of the abstract model used by the system.

The results of the expert's analysis showed the rewritten abstracts to be more informative in the sense that they contain more factual information than the original ones. However, they cannot be classified as "better quality" abstracts, as other kinds of writing problems still remain. The system focuses only on the rhetorical structure and there are other quality factors involved in the writing task, such as phrasing, grammar usage, register, etc. Nevertheless, the experiment showed that SciPo's critiquing tool offers potentially useful guidance towards more informative and genre-compliant abstracts.

5. Conclusions

We have reported on the porting of Argumentative Zoning from English to Portuguese. The features that were mostly affected by this porting were the syntactic ones: *Tense*, *Modal* and *Voice*, and also the *Formulaic* feature. Regarding the classification task (i.e. to assign one of the seven target categories to each sentence in our abstract corpus), we reported here the results of three experiments: (1) agreement results for human annotation, (2) intrinsic evaluation of automatic annotation and (3) intrinsic evaluation of automatic annotation taking context information into account. Our results are similar to Teufel and Moens's original results for English and they are very encouraging, particularly as the largest part of the porting could be performed in a matter of weeks.

The framework in which we use Argumentative Zoning is that of an automatic critiquing tool that is part of a bigger system for academic writing support in Portuguese, named SciPo. Being able to automatically determine the rhetorical status of a sentence put us in a position to implement a fully automatic critiquer, in addition to the currently implemented guided writing assistance. We reported an initial evaluation of the critiquing tool, which showed that Argumentative Zoning, although with some limitations, is suitable for this kind of application.

6. Acknowledgements

We would like to thank the financial support from the Brazilian funding agencies CAPES and FAPESP. We also want to thank Jorge Marques Pelizzoni and Lucas Antiqueira for their invaluable help implementing AZPort and SciPo.

7. Bibliography

Aires, R. V. X., Aluísio, S. M., Kuhn, D. C. S., Andreeta, M. L. B. and Oliveira Jr., O. N. (2000) Combining Multiple Classifiers to Improve Part of Speech Tagging: A Case Study for Brazilian Portuguese. In *Proceeding of SBIA 2000*. Atibaia, SP, Brazil.

Aluísio, S.M., Barcelos, I., Sampaio, J. and Oliveira Jr., O. (2001) How to learn the many unwritten "Rules of the Game" of the Academic Discourse: A hybrid Approach based on Critiques and Cases. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*. 257-260. Madison/Wisconsin.

Aluísio, S.M. and Oliveira Jr., O.N. (1996) A Detailed Schematic Structure of Research Papers Introductions: An Application in Support-Writing Tools. *Revista de la Sociedad Espanyola para el Procesamiento del Lenguaje Natural*, 19, 141-147.

Anthony, L. and Lashkia, G.V. (2003) Mover: A Machine Learning Tool to Assist in the Reading and Writing of Technical Papers. *IEEE Transactions on Professional Communication*, 46 (3), 185-193.

Broady, E. and Shurville, S. (2000) Developing Academic Writer: Designing a Writing Environment for Novice Academic Writers. In E. Broady (Ed.) *Second Language Writing in a Computer Environment*. 131-151. CILT, London.

Burstein, J., Marcu, D. and Knight, K. (2003) Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems: Special Issue on Natural Language Processing*, 18 (1), 32-39.

Feltrim, V., Aluísio, S. and Nunes, M.G.V. (2003) Analysis of the rhetorical structure of computer science abstracts in Portuguese. In *Proceedings of the Corpus Linguistics 2003*, Dawn Archer, Paul Rayson, Andrew Wilson and Tony McEnery (eds.), UCREL Technical Papers, Vol. 16, Part 1, Special Issue (2003) 212-218.

Kriegsman, M. and Barletta, R. (1993) Building a Case-based Help Desk Application. *IEEE Expert*, December, 18-26.

Liddy, E.D. (1991) The Discourse-Level Structure of Empirical Abstracts: An Exploratory Study. *Information Processing & Management*, 27 (1), 55-81.

Narita, M. (2000) Corpus-based English Language Assistant to Japanese Software Engineers. In *Proceedings of MT-2000 Machine Translation and Multilingual Applications in the New Millennium*. 24-1 – 24-8.

Santos M. (1996) The textual organisation of research paper abstracts. *Text*, 16 (4), 481-499.

Sharples, M., Goodlet, J. and Clutterbuck, A. (1994) A comparison of algorithms for hypertext notes network linearization. *International Journal of Human-Computer Studies*, 40 (4), 727-752.

Sharples, M. and Pemberton, L. (1992) Representing writing: external representations and the writing process. In P.O. Holt and N. Williams (Eds.) *Computers and Writing: State of the Art*. 319-336. Intellect, Oxford.

Siegel, S. and Castellan, N. (1988) *Nonparametric Statistics for the Behavioral Sciences,* McGraw-Hill.

Swales, J.M. (1990) *Genre Analysis: English in Academic and Research Settings,* Cambridge University Press. Cambridge, UK.

Teufel, S. and Moens, M. (2002) Summarising Scientific Articles – Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28 (4), 409-446.

Teufel, S. and Moens, M. (2000) What's yours and what's mine: Determining Intellectual Attribution in Scientific Text. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong.

Teufel, S., Carletta, J. and Moens, M. (1999) An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the Ninth Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*, 110-117.

Weissberg, R. and Buker, S. (1990) *Writing up Research: Experimental Research Report Writing for Students of English*, Prentice Hall.

Witten, I. and Frank, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann.