Self- or Pre-Tuning?
Deep linguistic processing of language variants

António Branco
Universidade de Lisboa
Antonio.Branco@di.fc.ul.pt

Francisco Costa
Universidade de Lisboa
fcosta@di.fc.ul.pt

Abstract

This paper proposes a design strategy for deep language processing grammars to appropriately handle language variants. It allows a grammar to be restricted as to what language variant it is tuned to, but also to detect the variant a given input pertains to. This is evaluated and compared to results obtained with an alternative strategy by which the relevant variant is detected with current language identification methods in a preprocessing step.

1 Introduction

This paper addresses the issue of handling different variants of a given language by a deep language processing grammar for that language.

In the benefit of generalization and grammar writing economy, it is desirable that a grammar can handle language variants – that share most grammatical structures and lexicon – in order to avoid endless multiplication of individual grammars, motivated by inessential differences.

From the viewpoint of analysis, however, increased variant coverage typically opens the way to increased spurious overgeneration. Consequently, the ability for the grammar to be tuned to the relevant dialect of the input is important to control overgeneration arising from its flexibility.

Control on what is generated is also desirable. In general one wants to be able to parse as much variants as possible, but at the same time be selective in generation, by consistently generating only in a given selected variant.

Closely related to the setting issue (addressed in the next Section 2) is the tuning issue: if a system can be restricted to a particular variety, what is the best way to detect the variety of the input? We discuss two approaches to this issue.

One of them consists in using pre-processing components that can detect the language variety at stake. This pre-tuning approach explores the hypothesis that methods developed for language identification can be used also to detect language variants (Section 5).

The other approach is to have the computational grammar prepared for self-tuning to the language variant of the input in the course of processing that input (Section 4).

We evaluate the two approaches and compare them (last Section 6).

2 Variant-sensitive Grammar

In this Section, we discuss the design options for a deep linguistic processing grammar allowing for its appropriate tuning to different language variants. For the sake of concreteness of the discussion, we assume the HPSG framework (Pollard and Sag, 1994) and a grammar that handles two close variants of the same language, European and Brazilian Portuguese. These assumptions are merely instrumental, and the results obtained can be easily extended to other languages and variants, and to other grammatical frameworks for deep linguistic processing.

A stretch of text from a language L can display grammatical features common to all variants of L, or contain a construction that pertains to some or only one of its variants. Hence, undesirable overgeneration due to the grammar readiness to cope with all language variants can

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 57–64,
Ann Arbor, June 2005.

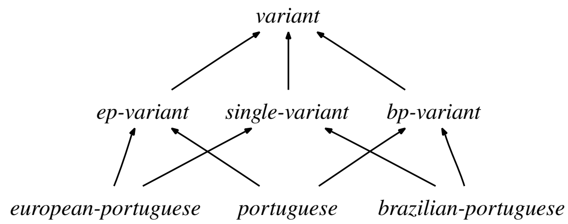

Figure 1: Type hierarchy under *variant*.

be put in check by restricting the grammar to produce variant-"consistent" analyses. More precisely, if the input string contains an element that can only be found in variety v_1 and that input string yields ambiguity in a different stretch but only in varieties v_k other than v_1, this ambiguity will not give rise to multiple analyses if the grammar can be designed so that it can be constrained to accept strings with marked elements of at most one variety, v_1.

The approach we propose seeks to implement this mode of operation in analysis, with the important effect of permitting also to control the variant under which generation should be performed. It relies on the use of a feature VARIANT to model variation. This feature is appropriate for all signs and declared to be of type *variant*. Given the working language variants assumed here, its values are presented in Figure 1.

This attribute is constrained to take the appropriate value in lexical items and constructions specific to one of the two varieties. For example, a hypothetical lexical entry for the lexical item *autocarro* (*bus*, exclusive to European Portuguese) would include the constraint that the attribute VARIANT has the value *ep-variant* and the corresponding Brazilian Portuguese entry for *ônibus* would constrain the same feature to bear the value *bp-variant*. The only two types that are used to mark signs are *ep-variant* and *bp-variant*. The remaining types presented in Figure 1 are used to constrain grammar behavior, as explained below.

Lexical items are not the only elements that can have marked values in the VARIANT feature. Lexical and syntax rules can have them, too. Such constraints model constructions that markedly pertain to one of the dialects.

Feature VARIANT is structure-shared among all signs comprised in a full parse tree. This is achieved by having all lexical or syntactic rules unifying their VARIANT feature with the VARIANT feature of their daughters.

If two signs (e.g. from lexical items and syntax rules) in the same parse tree have different values for feature VARIANT (one has *ep-variant* and the other *bp-variant*), they will unify to *portuguese*, as can be seen from Figure 1. This type means that lexical items or constructions specific to two different varieties are used together. Furthermore, since this feature is shared among all signs, it will be visible everywhere, for instance in the root node.

It is possible to constrain feature VARIANT in the root condition of the grammar so that the grammar works in a variant-"consistent" fashion: this feature just has to be constrained to be of type *single-variant* (in root nodes) and the grammar will accept either European Portuguese or Brazilian Portuguese. Furthermore, in the non natural condition where the input string bears marked properties of both variants, that string will receive no analysis: feature VARIANT will have the value *portuguese* in this case, and there is no unifier for *portuguese* and *single-variant*.

If this feature is constrained to be of type *european-portuguese* in the root node, the grammar will not accept any sentence with features of Brazilian Portuguese, since they will be marked to have a VARIANT of type *bp-variant*, which is incompatible with *european-portuguese*. It is also possible to have the grammar reject European Portuguese (using type *brazilian-portuguese*) or to ignore variation completely by not constraining this feature in the start symbol.

With this grammar design it is thus possible to control beforehand the mode of operation for the grammar, either for it to handle only one variant or several. But it is also possible to use the grammar to detect to which variety input happens to belong. This self-tuning of the grammar to the relevant variant is done by parsing that input and placing no constraint on feature VARIANT of root nodes, and then reading the value of attribute VARIANT from the resulting feature structure: values *ep-variant* and *bp-variant* result from parsing text with properties specific to European Portuguese or Brazilian Portuguese respectively; value *variant* indicates that no marked elements were detected and the text can be from both variants. Also here where the language variant of the input is detected by the grammar, the desired variant-"consistent" behavior of the grammar is enforced.

If the input can be known to be specifically European or Brazilian Portuguese before it is parsed, the constraints on feature VARIANT can be set accordingly to improve efficiency: When parsing text known to be European Portuguese, there is no need to explore analyses that are markedly Brazilian Portuguese, for instance.

It is thus important to discuss what methods for language variant detection can be put in place that support a possible pre-processing step aimed at pre-tuning the grammar for the relevant variant of the input. It is also important to gain insight on the quality of the performance of this method and on how the performance of this pre-tuning setup compares with the self-tuning approach. This is addressed in the next Sections.

3 Experimental setup

Before reporting on the results obtained with the experiments on the performance of the two approaches (self- and pre-tuning), it is important to introduce the experimental conditions under which such exercises were conducted.

3.1 Data

To experiment with any of these two approaches to variant-tuning, two corpora of newspaper text were used, CETEMPublico (204M tokens) and CETENFolha (32M tokens). The first contains text from the European newspaper *O Público*, and the latter from the South American *Folha de São Paulo*. These corpora are only minimally annotated (paragraph and sentence boundaries, *inter alia*), but are very large.

Some preprocessing was carried out: XML-like tags, like the `<s>` and `</s>` tags marking sentence boundaries, were removed and each individual sentence was put in a single line.

Some heuristics were also employed to remove loose lines (parts of lists, etc.) so that only lines ending in `.`, `!` and `?`, and containing more than 5 tokens (whitespace delimited) were considered. Other character sequences that were judged irrelevant and potential misguiders for the purpose at hand were normalized: URLs were replaced by the sequence `URL`, e-mail addresses by `MAIL`, hours and dates by `HORA` and `DATA`, etc. Names at the beginning of lines indicating speaker (in an interview, for instance) were removed, since they are frequent and the grammar that will be used is not intended to parse name plus sentence strings.

The remaining lines were ordered by length in terms of words and the smallest 200K lines from each of the two corpora were selected. Small lines were preferred as they are more likely to receive an analysis by the grammar.

Given the methods we will be employing for pre-tuning reportedly perform well even with small training sets (Section 5), only a modest portion of text from these corpora was needed.

In the benefit of comparability of the two approaches for grammar tuning, it is important that all the lines in the working data are parsable by the grammar. Otherwise, even if in the pre-tuning approach the pre-processor gets the classification right for non parsable sentences, this will be of no use since the grammar will not produce any result out of that. 90K lines of text were thus randomly selected from each corpus and checked as to whether they could be parsed by the grammar. 25K of parsable lines of the American corpus and 21K of parsable lines of the European corpus were obtained (46K lines out of 180K, representing 26% rate of parsability for the grammar used – more details on this grammar in the next Section).

It is worth noting that the use of two corpora, one from an European newspaper and the other from an American newspaper, without further annotation, does not allow their appropriate use in the present set of experiments. The reason is that if a sentence is found in the European corpus, one can have almost absolute certainty that it is possible in European Portuguese, but one does not know if it is Brazilian Portuguese, too. The same is true of any sentences in the American corpus — it can also be a sentence of European Portuguese in case it only contains words and structures common to both variants.

In order to prepare the data, a native speaker of European Portuguese was asked to manually decide from sentences found in the American corpus whether they are markedly Brazilian Portuguese. Conversely, a Brazilian informant detected markedly European Portuguese sentences from the European corpus.

From these parsed lines we drew around 1800 random lines of text from each corpus, and had them annotated. The lines coming from the American corpus were annotated for whether they are markedly Brazilian Portuguese, and vice-versa for the other corpus. Thus a three-way classification is obtained: any sentence was classified as being markedly Brazilian Portuguese, European Portuguese or common to both variants.

The large majority of the sentences were judged to be possible in both European and Brazilian Portuguese. 16% of the sentences in the European corpus were considered not belonging to Brazilian Portuguese, and 21% of the sentences in the American corpus were judged as not being European Portuguese.[1] Overall, 81% of the text was common to both varieties.

10KB of text from each one of the three classes were obtained. 140 lines, approximately 5KB, were reserved for training and another 140 for test. In total, the 30 K corpus included 116, 170, 493 and 41 sentence tokens for, respectively, 8, 7, 6 and 5 word length sentence types.

3.2 Variation

These training corpora were submitted to manual inspection in order to identify and quantify the sources of variant specificity. This is important to help interpret the experimental results and to gain insight on the current coverage of the grammar used in the experiment.

This analysis was performed over the 140 lines selected as markedly Brazilian Portuguese, and assumed that the sources of variant specificity should have broadly the same distribution in the other 140K lines markedly European Portuguese.

1. Mere orthographic differences (24%) e.g. *ação* vs. *acção* (*action*)

2. Phonetic variants reflected in orthography (9.3%) e.g. *irônico* vs. *irónico* (*ironic*)

3. Lexical differences (26.9% of differences)
   (a) Different form, same meaning (22.5%) e.g. *time* vs. *equipa* (*team*)
   (b) Same form, different meaning (4.4%) e.g. *policial* (*policeman/criminal novel*

4. Syntactic differences (39.7%)
   (a) Possessives w/out articles (12.2%)
   (b) In subcategorization frames (9.8%)
   (c) Clitic placement (6.4%)
   (d) Singular bare NPs (5.4%)
   (e) In subcat and word sense (1.9%)
   (f) Universal *todo* + article (0.9%)
   (g) Contractions of Prep+article (0.9%)
   (h) Questions w/out SV inversion (0.9%)
   (i) Postverbal negation (0.5%)
   (j) other (0.5%)

About 1/3 of the differences found would disappear if a unified orthography was adopted. Differences that are reflected in spelling can be modeled via multiple lexical entries, with constraints on feature VARIANT reflecting the variety in which the item with that spelling is used.

Interestingly, 40% of the differences are syntactic in nature. These cases are expected to be more difficult to detect with stochastic approaches than with a grammar.

4 Self-tuning

4.1 Grammar and baseline

The experiments on the self-tuning approach were carried out with a computational grammar for Portuguese developed with the LKB platform (Copestake, 2002) that uses MRS for semantic representation (Copestake et al., 2001) (Branco and Costa, 2005). At the time of the experiments reported here, this grammar was of modest size. In terms of linguistic phenomena, it covered basic declarative sentential structures and basic phrase structure of all categories, with a fully detailed account of the structure of NPs. It contained 42 syntax rules, 37 lexical rules (mostly inflectional) and a total of 2988 types, with 417 types for lexical entries. There were 2630 hand-built lexical entries, mostly nouns, with 1000 entries. It was coupled with a POS tagger for Portuguese, with 97% accuracy (Branco and Silva, 2004).

[1] A hypothetical explanation for this asymmetry (16% vs. 21%) is that one of the most pervasive differences between European and Brazilian Portuguese, clitic placement, is attenuated in writing: Brazilian text often displays word order between clitic and verb similar to European Portuguese, and different from oral Brazilian Portuguese. Therefore, European text displaying European clitic order tends not be seen as markedly European. In fact, we looked at the European sentences with clitic placement characteristic of European Portuguese that were judged possible in Brazilian Portuguese. If they were included in the markedly European sentences, 23% of the European text would be unacceptable Brazilian Portuguese, a number closer to the 21% sentences judged to be exclusively Brazilian Portuguese in the American corpus.

In terms of the sources of variant specificity identified above, this grammar was specifically designed to handle the co-occurrence of prenominal possessives and determiners and most of the syntactic constructions related to clitic-verb order. As revealed by the study of the training corpus, these constructions are responsible for almost 20% of marked sentences.

The lexicon contained lexical items markedly European Portuguese and markedly Brazilian Portuguese. These were taken from the Portuguese Wiktionary, where this information is available. Leaving aside the very infrequent items, around 740 marked lexical items were coded. Items that are variant specific found in the training corpora (80 more) were also entered in the lexicon.

These items, markedly belonging to one variant, were declined into their inflected forms and the resulting set Lex_{bsl} was used in the following baseline for dialect tuning: for a sentence s and N_{ep}, resp. N_{bp}, the number of tokens of items in Lex_{bsl} markedly European, resp. Brazilian Portuguese, occurring in s, s is tagged as European Portuguese if $N_{ep} > N_{bp}$, or vice-versa, or else, "common" Portuguese if $N_{ep} = N_{bp} = 0$.

Known class	*Predicted class* EP	BP	Common	Recall
EP	45	0	95	0.32
BP	3	45	92	0.32
Common	4	4	132	0.94
Precision	0.87	0.98	0.41	

Table 1: Baseline: Confusion matrix.

For this baseline, the figure of 0.53 of overall accuracy was obtained, detailed in Table 1.[2]

4.2 Results with self-tuning

The results obtained for the self-tuning mode of operation are presented in Table 2.[3] When the grammar produced multiple analyses for a given sentence, that sentence was classified as markedly European, resp. Brazilian, Portuguese if all the parses produced VARIANT with type *ep-variant*, resp. *bp-variant*. In all other cases, the sentence would be classified as common to both variants.

Known class	*Predicted class* EP	BP	Common	Recall
EP	53	1	86	0.38
BP	6	61	73	0.44
Common	14	1	125	0.89
Precision	0.73	0.97	0.44	

Table 2: Self-tuning: Confusion matrix.

Every sentence in the test data was classified, and the figure of 0.57 was obtained for overall accuracy. The analysis of errors shows that the sentence belonging to Brazilian Portuguese or to "common" Portuguese wrongly classified as European Portuguese contain clitics following the European Portuguese syntax, and some misspellings conforming to the European Portuguese orthography.

5 Pre-tuning

5.1 Language Detection Methods

Methods have been developed to detect the language a given text is written in. They have also been used to discriminate varieties of the same language, although less often. (Lins and Gonçalves, 2004) look up words in dictionaries to discriminate among languages, and (Oakes, 2003) runs stochastic tests on token frequencies, like the chi-square test, in order to differentiate between European and American English.

Many methods are based on frequency of byte n-grams in text because they can simultaneously detect language and character encoding (Li and Momoi, 2001), and can reliably classify short portions of text. They have been applied in web browsers (to identify character encodings) and information retrieval systems.

We are going to focus on methods based on character n-grams. Because all information used for classification is taken from characters, and they can be found in text in much larger quantities than words or phrases, problems of scarcity of data are attenuated. Besides, training data can also be easily found in large amounts because corpora do not need to be annotated (it is

[2] Naturally, extending the operation of this baseline method beyond the terms of comparability with grammars that handle each sentence at a time, namely by increasingly extending the number of sentences in the stretch of text being classified, will virtually lead it to reach optimal accuracy.

[3] These figures concern the test corpus, with the three conditions represented by 1/3 of the sentences, which are all parsable. Hence, actual recall over a naturally occurring text is expected to be lower. Using the estimate that only 26% of input receives a parse, that figure for recall would lie somewhere around 0.15 (= 0.57 x 0.26).

only necessary to know the language they belong to). More importantly, methods based on character n-grams can reliably classify small portions of text. The literature on automatic language identification mentions training corpora as small as 2K producing classifiers that perform with almost perfect accuracy for test strings as little as 500 Bytes (Dunning, 1994) and considering several languages. With more training data (20K-50K of text), similar quality can be achieved for smaller test strings (Prager, 1999).

Many n-gram based methods have been explored besides the one we opted for.[4] Many can achieve perfect or nearly perfect classification with small training corpora on small texts. In previous work (Branco and Costa, 2007), we did a comparative study on two classifiers that use approaches very well understood in language processing and information retrieval, namely Vector Space and Bayesian models. We retain here the latter as this one scored comparatively better for the current purposes.

In order to know which language $L_i \in L$ generated string s, Bayesian methods can be used to calculate the probabilities $P(s|L_i)$ of string s appearing in language L_i for all $L_i \in L$, the considered language set, and decide for the language with the highest score (Dunning, 1994). That is, in order to compute $P(L_i|s)$, we only compute $P(s|L_i)$. The Bayes rule allows us to cast the problem in terms of $\frac{P(s|L_i)P(L_i)}{P(s)}$, but as is standard practice, the denominator is dropped since we are only interested here in getting the highest probability, not its exact value. The prior $P(L_i)$ is also ignored, corresponding to the simplifying assumption that all languages are equally probable for the operation of the classifier. The way $P(s|L_i)$ is calculated is also the standard way to do it, namely assuming independence and just multiplying the probabilities of character c_i given the preceding *n-1* characters (using n-grams), for all characters in the input (estimated from n-gram counts in the training set).

For our experiments, we implemented the algorithm described in (Dunning, 1994). Other common strategies were also used, like prepending $n-1$ special characters to the input string to harmonize calculations, summing logs of probabilities instead of multiplying them to avoid underflow errors, and using Laplace smoothing to reserve probability mass to events not seen in training.

[4]See (Sibun and Reynar, 1996) and (Hughes et al., 2006) for surveys.

5.2 Calibrating the implementation

5.2.1 Detection of languages

First of all, we want to check that the language identification methods we are using, and have implemented, are in fact reliable to identify different languages. Hence, we run the classifier on three languages showing strikingly different characters and character sequences. This is a deliberately easy test to get insight into the appropriate setting of the two parameters at stake here, size of of the n-gram in the training phase, and size of the input in the running phase.

For this test, we used the Universal Declaration of Human Rights texts.The languages used were Finnish, Portuguese and Welsh.[5]

Several tests were conducted, splitting the test data in chunks 1, 5, 10 and 20 lines long. The classifier obtained perfect accuracy on all test conditions (all chunk sizes), for all values of n between 1 and 7 (inclusively). For $n = 8$ and $n = 9$ there were errors only when classifying 1 line long items.

The average line length for the test corpora was 138 characters for Finnish, 141 for Portuguese and 121 for Welsh (133 overall). In the corpora we will be using in the following experiments, average line length is much lower (around 40 characters per line). To become closer to our experimental conditions, we also evaluated this classifiers with the same test corpora, but truncated each line beyond the first 50 characters, yielding test corpora with an average line length around 38 characters (since some were smaller than that). The results are similar. The Bayesian classifier performed with less than perfect accuracy also with $n = 7$ when classifying 1 line at a time.

Our classifier was thus performing well at discriminating languages with short values of n, and can classify short bits of text, even with incomplete words.

[5]The Preamble and Articles 1–19 were used for training (8.1K of Finnish, 6.9K of Portuguese, and 6.1K of Welsh), and Articles 20–30 for testing (4.6K of Finnish, 4.7K of Portuguese, and 4.0K of Welsh).

5.2.2 Detection of originating corpus

In order to study its suitability to discriminate also the two Portuguese variants, we experimented our implementation of the Bayesian classifiers on 200K lines of text from each of the two corpora. We randomly chose 20K lines for testing and the remaining 180K for training. A classification is considered correct if the classifier can guess the newspaper the text was taken from.

The average line length of the test sentences is 43 characters. Several input lengths were tried out by dividing the test data into various sets with varying size. Table 3 summarizes the results obtained.

	Length of Test Item			
	1 line	5 lines	10 lines	20 lines
$n = 2$	0.84	0.99	**1**	**1**
$n = 3$	**0.96**	0.99	**1**	**1**
$n = 4$	**0.96**	**1**	**1**	**1**
$n = 5$	0.94	**1**	**1**	**1**
$n = 6$	0.92	0.99	**1**	**1**
$n = 7$	0.89	0.98	0.99	**1**

Table 3: Originating corpora: Accuracy

The accuracy of the classifier is surprisingly high given that the sentences that cannot be attributed to a single variety are estimated to be around 81%.

5.2.3 Scaling down the training data

A final check was made with the classifier to gain further insight on the comparability of the results obtained under the two tuning approaches. It was trained on the data prepared for the actual experiment, made of the 10K with lines that have the shortest length and are parsable, but using only the markedly European and Brazilian Portuguese data (leaving aside the sentences judged to be common to both). This way the two setups can be compared, since in the test of the Subsection just above much more data was available for training.

Results are in Table 4. As expected, with a much smaller amount of training data there is an overall drop in the accuracy, with a noticed bias at classifying items as European Portuguese. The performance of the classifier degrades with larger values of n. Nevertheless, the classifier is still very good with bigrams, with an

	Length of Test Item			
	1 line	5 lines	10 lines	20 lines
$n = 2$	**0.86**	**0.98**	**0.96**	**1**
$n = 3$	0.82	0.73	0.64	0.5
$n = 4$	0.68	0.55	0.5	0.5

Table 4: Two-way classification: Accuracy

almost optimal performance, only slightly worse than the one observed in the previous Subsection, when it was trained with more data.

From these preliminary tests, we learned that we could expect a quasi optimal performance of the classifier we implemented to act as a preprocessor in the pre-tuning approach, when $n = 2$ and it is run under conditions very close to the ones it will encounter in the actual experiment aimed at comparing the two tuning approaches.

5.3 Results with pre-tuning

In the final experiment, the classifier should discriminate between three classes, deciding whether the input is either specifically European or Brazilian Portuguese, or else whether it belongs to both variants. It was trained over the 15K tokens/420 lines of training data, and tested over the held out test data of identical size.

	Length of Test Item			
	1 line	5 lines	10 lines	20 lines
$n = 2$	**0.59**	**0.67**	**0.76**	**0.76**
$n = 3$	0.55	0.52	0.45	0.33
$n = 4$	0.48	0.39	0.33	0.33

Table 5: Three-way classification: Accuracy

The results are in Table 5. As expected, the classifier based in bigrams has the best performance for every size of the input, which improves from 0.59 to 0.76 as the size of the input gets from 1 line to 20 lines.

6 Discussion and conclusions

From the results above for pre-tuning, it is the value 0.59, obtained for 1 line of input, that can be put on a par with the value of 0.57 obtained for self-tuning — both of them to be appreciated against the baseline of 0.53.

Interestingly, the performance of both approaches are quite similar, and quite encouraging given the limitations under which the present pilot exercise was executed. But this is also the reason why they should be considered with the appropriate *grano salis*.

Note that there is much room for improvement in both approaches. From the several sources of variant specificity, the grammar used was prepared to cope only with grammatical constructs that are responsible for at most 20% of them. Also the lexicon, that included a little more than 800 variant-distinctive items, can be largely improved.

As to the classifier used for pre-tuning, it implements methods that may achieve optimal accuracy with training data sets of modest size but that need to be nevertheless larger than the very scarce 15K tokens used this time. Using backoff and interpolation will help to improve as well.

Some features potentially distinguish, however, the pre-tuning based on Bayesian classifier from the self-tuning by the grammar.

Language detection methods are easy to scale up with respect to the number of variants used. In contrast, the size of the type hierarchy under *variant* is exponential on the number of language variants if all combinations of variants are taken into account, as it seems reasonable to do.

N-grams based methods are efficient and can be very accurate. On the other hand, like any stochastic method, they are sensitive to training data and tend to be much more affected than the grammar in self-tuning by a change of text domain. Also in dialogue settings with turns from different language variants, hence with small lengths of texts available to classify and successive alternation between language variants, n-grams are likely to show less advantage than self-tuning by fully fledged grammars.

These are issues over which more acute insight will be gained in future work, which will seek to improve the contributions put forward in the present paper.

Summing up, a major contribution of the present paper is a design strategy for type-feature grammars that allows them to be appropriately set to the specific language variant of a given input. Concomitantly, this design allows the grammars either to be pre-tuned or to self-tune to that dialect – which, to the best of our knowledge, consists in a new kind of approach to handling language variation in deep processing.

In addition, we undertook a pilot experiment which can be taken as setting the basis for a methodology to comparatively assess the performance of these different tuning approaches and their future improvements.

References

António Branco and Francisco Costa. 2005. LX-GRAM – deep linguistic processing of Portuguese with HSPG. Technical report, Dept. of Informatics, University of Lisbon.

António Branco and Francisco Costa. 2007. Handling language variation in deep processing. In *Proc. CLIN2007*.

António Branco and João Silva. 2004. Evaluating solutions for the rapid development of state-of-the-art POS taggers for Portuguese. In *Proc. LREC2004*.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2001. Minimal Recursion Semantics: An introduction. *Language and Computation*, 3.

Ann Copestake. 2002. *Implementing typed feature structure grammars*. CSLI.

Ted Dunning. 1994. Statistical identification of language. Technical Report MCCS-94-273, Computing Research Lab, New Mexico State Univ.

Baden Hughes, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay. 2006. Reconsidering language identification for written language resources. In *Proc. LREC2006*.

Shanjian Li and Katsuhiko Momoi. 2001. A composite approach to language/encoding detection. In *Proc. 19th International Unicode Conference*.

Rafael Lins and Paulo Gonçalves. 2004. Automatic language identification of written texts. In *Proc. 2004 ACM Symposium on Applied Computing*.

Michael P. Oakes. 2003. Text categorization: Automatic discrimination between US and UK English using the chi-square test and high ratio pairs. *Research in Language*, 1.

Carl Pollard and Ivan Sag. 1994. *Head-driven phrase structure grammar*. CSLI.

John M. Prager. 1999. Linguini: Language identification for multilingual documents. *Journal of Management Information Systems*, 16(3).

Penelope Sibun and Jeffrey C. Reynar. 1996. Language identification: Examining the issues. In *5th Symposium on Document Analysis and IR*.