Robustness and customisation in an analyser/lemmatiser for Basque

Alegria I., Aranzabe M., Ezeiza A., Ezeiza N., Urizar R.

Informatika Fakultatea
649 P.K. E-20080 Donostia. Basque Country.
{i.alegria, jipecran}@si.ehu.es

Abstract

This paper describes the work carried out to improve the robustness of the morphological analyser/generator for Basque which can be adapted to several domains and variants of the language. This analyser is used as a lemmatiser in several IR applications such as an Intranet search engine.

We present an enhanced analyser that deals not only with standard words but also with linguistic variants (including dialectal variants and competence errors) and words, whose lemmas are not included in the lexicon, by relaxing the constraints of the standard analyser. In addition to this, a user's lexicon can be added to the system in order to customise the tool. This user's lexicon can be obtained by means of a semiautomatic process.

1. Introduction

The starting point of this research is a general morphological analyser/generator described in (Alegria et al., 1996), which reported 95% of coverage. This poor result was due (at least partially) to the recent standardisation and the widespread dialectal use of Basque.

Although in some systems lemmas corresponding to unknown words are included in the main lexicon in a previous step, this solution is not satisfactory if we want to build a flexible system. We decided that it was necessary to manage a user's lexicon, for linguistic variants and forms whose lemmas were not in the lexicon, if we wanted to develop a comprehensive or adapted analyser.

However, the enhancement of coverage leads, in some cases, to produce overgeneration, and, consequently, to increase ambiguity. Although this ambiguity is not real, it causes poor results (lower precision) in applications based on morphology or lemmatisation. Another important issue was the improvement of precision. We studied the results of the analyser and saw that most errors (50%-75%) were made when dealing with proper names. Therefore, we propose some solutions to avoid about 50% of the errors.

2. Architecture of the morphological analyser

Morfeus is a robust morphological analyser for Basque. It is a basic tool for current and future work on NLP of Basque. Some of the tools based on it are a tagger (Ezeiza et al., 1998), an Intranet search engine (Aizpurua et al., 2000) and an assistant for verse making (et al., 2001).

The analyser is based on the two-level formalism. The two-level model of computational morphology was proposed by Koskenniemi (Koskenniemi, 1983) and has had widespread acceptance due mostly to its general applicability, declarativeness of rules and clear separation of linguistic knowledge and program.

This tool is implemented using lexical transducers. A lexical transducer (Karttunen, 1994) is a finite-state automaton that maps inflected surface forms to lexical forms, and can be considered an evolution of the two-level morphology. The tool used for the implementation is the *fst* library of *Inxight*[1] (Karttunen and Bessley, 1992; Karttunen, 1993; Karttunen et al., 1996). A detailed description of the transducers can be found in (Alegria et al., 2001).

We have defined the architecture of the analyser using three main modules (Schiller (Schiller, 1996) and others propose only two levels):

1. The standard analyser that uses a general lexicon and a user's lexicons. This module is able to analyse/ generate standard language word-forms. In our applications for Basque we defined about 75,000 entries in the general lexicon, more than 130 patterns of morphotactics and two rule systems in cascade, the first one for long-distance dependencies among morphemes and the second for morphophonological changes. The three elements are compiled together in the standard transducer. To deal with the user's lexicon the general transducer described below is used.
2. The analysis and normalization of linguistic variants (dialectal uses and competence errors). Due to non-standard or dialectal uses of the language and competence errors, the standard morphology is not enough to offer good results when analysing real text corpora. This problem becomes critical in languages like Basque in which standardisation is in process and dialectal forms are still of widespread use. For this process the standard transducer is extended with new lexical entries and phonological rules producing the *enhanced transducer*.
3. The guesser or analyser of words without lemmas in the lexicons. In this case the standard transducer is simplified removing the lexical entries in open categories (nouns, adjectives, verbs, …), which constitute the vast majority of the entries, and is substituted by a general automata to describe any combination of characters. So, the *general transducer* is produced combining this general set of lemmas with affixes related to open categories and general rules.

[1] Inxight Software, Inc., a Xerox Enterprise Company (www.inxight.com)

The analyser of non-standard words (steps 2 and 3) may sometimes produce overgeneration, and it is important to reduce this ambiguity as soon as possible.

3. Customizing the analyser

In order to deal with unknown words, a general transducer has been designed to relax the need of lemmas in the lexicon. This transducer was initially (Alegria et al., 1997) based on an idea used in a speech synthesis system (Black et al., 1991) but it has been now simplified. Daciuk (Daciuk, 2000) proposes a similar way when he describes the *guessing automaton*, but the construction of our automaton is simpler.

The new transducer is the standard one modified in this way: the lexicon is reduced to affixes corresponding to open categories and generic lemmas for each open class, while standard rules remain. There are seven open classes and the most important ones are: common nouns, personal names, place nouns, adjectives and lexical verbs. Grammatical categories and semantic ones (personal names or place names) are separated because they have different declension.

So, the standard rule-system is composed of a mini-lexicon where the generic lemmas are obtained as a result of combining alphabetical characters and can be expressed in the lexicon as a cyclic sublexicon with the set of letters (some constraints are used with capital/non-capital letters according to the part of speech). In fig. 1 the graph corresponding to the mini-lexicon is shown.

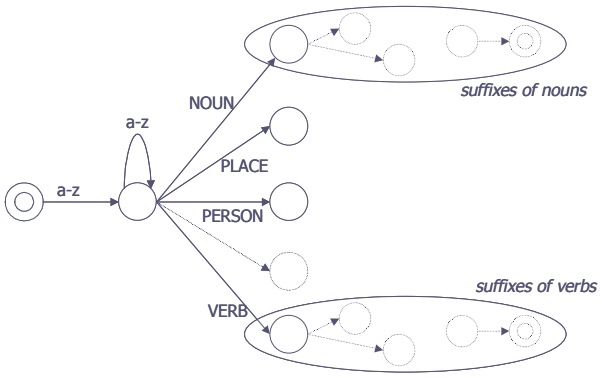

Figure 1. Simplified graph of the mini-lexicon

This transducer is used in two steps of the analysis:

1. in the standard analysis, in order to analyse declension and derivation of lemmas in the user's lexicon.
2. in the analysis without lexicon (called *guesser* in taggers).

The user's lexicon is composed of a list of lemmas along with their parts of speech defined by the users. The general transducer suggests possible interpretations of the word, and these lemmas are searched in the user's lexicon. When any lemma and class given by the general transducer matches the information on the user's lexicon, the analyser selects the corresponding interpretation and gives it as a result.

So, the user's lexicon is an editable resource which can be inferred from corpora or be managed on-line by the user. The use of this lexicon combined with the general transducer allows to customise the applications and it has been included successfully in three tools:

1. A spelling corrector for Basque (Aldezabal et al., 1999) in which for each lemma included in the user's lexicon any inflected form or derivative is accepted.
2. An Intranet search engine (Aizpurua et al., 2000) in which lemmatisation plays an important role and which can be customised when adapted to a special domain. In this case a semiautomatic process is carried out. First, the whole analyser (in the three steps above mentioned) is used to analyse a big corpus and the possible lemmas obtained by the guesser. After being sorted by frequency, they are presented to the user in order to include them in the user's lexicon[2]. The site www.zientzia.net, devoted to scientific documents, was built in this way.
3. A general part-of-speech tagger including customisation similar to the search engine.

4. Increasing coverage

The analyser was designed with the main objective of being robust, that is, capable of treating both standard and non-standard forms in real texts. For this reason, the morphological analyser has been extended in two ways:

1. The treatment of linguistic variants (dialectal variants and competence errors) (Aduriz *et al.*, 1994)
2. A two-level mechanism for lemmatisation without lexicon to deal with unknown words, which has been explained above

Important features of this design are homogeneity, modularity and reusability because the different steps are based on lexical transducers, far from *ad hoc* solutions, and these elements can be used in different tools. This could be considered a variant of constraint relaxation techniques used in syntax (Stede, 1992), where the first constraint demands standard language, the second one combines standard and linguistic variants, and the third step allows free lemmas in open categories. Only if the previous steps fail, the results of the next step are included in the output. Oflazer also uses relaxation techniques in morphology (Oflazer, 1996).

With this design the obtained coverage is 100% and precision over 99.5%. The ambiguity measures of the morphological analyser, taken from a balanced corpus of about 27,000 tokens and from a news collection of about 9,000, are shown in table 1. These measures have been obtained using all the morphological features.

Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token
66.95%	4.38	3.26

Table 1: Ambiguity measures[3]

However, sometimes overgeneration is produced in order to improve robustness. Overgeneration increases ambiguity but often this ambiguity is not real and causes poor results (low precision) in applications based on morphology such as spelling correction, morphological generation or tagging.

[2] At this moment it is a not friendly off-line process

[3] Ambiguity Rate: #ambiguous_token / #token; Interpretations per token: #analyses / #token; Interpretations per ambiguous token: #analyses_ambiguous_token / # ambiguous_token

	Distribution	Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token	Precision
standard	77.90%	80.73%	3.81	3.27	99.73%
variant	1.75%	80.53%	4.23	3.60	92.31%
unknown	2.65%	99.79%	18.05	18.01	98.12%
average	100.00%	66.95%	4.38	3.26	99.61%

Table 2: Ambiguity measures in the output of the analyser

	tokens	standard	variant	unknown	other[4]
corpus1	116,720	76.66%	1.02%	3.28%	19.04%
corpus2	1,288,257	78.44%	0.94%	3.80%	16.82%
corpus3	587,515	74.98%	2.03%	2.92%	20.07%
corpus4	33,232	77.32%	1.42%	4.92%	16.34%
corpus5	148,333	77.91%	1.01%	6.23%	14.85%
corpus6	29,939	60.54%	11.50%	7.90%	20.06%

Table 3: Distribution of tokens in different types of corpora

5. Decreasing ambiguity

The ambiguity for linguistic variants and unknown words is higher and the precision measures are poorer, but they form a small group of the input words (5%-10%) and the influence on average results is not significant.

The morphological analyser may sometimes overgenerate analyses of linguistic variants and unknown lemmas (table 2). Even if most words in texts are analysed in the first phase (see table 3), the small proportion of non-standard words constitutes a great amount of the superfluous interpretations. Yet, the rate of non-standard words varies depending on the type of corpus.

For instance, corpus3 is a balanced corpus with a high rate of standard Basque texts. On the contrary, corpus6 is a subset of texts from corpus3 written mainly in two dialects. Obviously, this corpus has a higher rate of non-standard uses. Corpus1 is a compilation of texts from the Web, and, generally, there is a trend to write these documents following standard rules of the language. Finally, corpus2, corpus4 and corpus5 are texts from the Basque newspaper *Euskaldunon Egunkaria*, and, even if the language variant used on them is standard, there is a relatively high amount of unknown words.

The treatment of non-standard words has been added to the previously developed analyser for two main reasons:

1. The average number of interpretations in non-standard words is significantly higher than in standard words (see table 2).
2. There could be multiple lemmas for the same or similar morphological analysis. This is a problem when we want to build a lemmatiser. For example, if *bitaminiko* (vitaminic) is not in the lexicon the results of the analysis of *bitaminikoaren* (from the vitaminic) as adjective can be multiple: *bitamini+ko+aren*, *bitaminiko+aren* and *bitaminikoaren*, but the only right analysis is the second one.

We think that it is important to reduce the ambiguity at this stage, so that the input of subsequent processes is more precise. But, we do not use information about surrounding words because a tagger will be used later. The process is limited to the word we want to treat, and we only need to know, in some cases, if the previous token was a full stop.

This module consists of different methods for linguistic variants and unknown words, because overgeneration is produced by different facts in each case, as will be described below.

5.1. Disambiguation of linguistic variants

In the case of linguistic variants a heuristic tries to select the lemma that is "nearest" to the standard one according to the number of non-standard morphemes and rules applied. It chooses the interpretation that has less non-standard uses for each POS tag.

For example, analysing the word-form *kaletikan* (dialectal form) two possible analyses are obtained: *kale+tik* (from the street) and *kala+tik* (from the cove). Both analyses have a non-standard morpheme (*-tikan*) but the first analysis is more probable because it applies no other transformation rule and to obtain the second one it has been necessary to apply another rule at the end of the lemma to transform *kale* into *kala*.

Thus, we must decide which of the analyses need to be selected or discarded based on the amount of transformation rules applied to obtain each analysis, but the enhanced transducer does not detail this information. The output of the enhanced transducer displays the normalised lemma/morphemes along with their corresponding morphological features. In the case of non-standard morphemes linked in the lexical database to their normalised form, the analysis details both normalised and variant morphemes.

Thus, the procedure uses these results to select the most probable lemmas for each POS tag. The results of applying this procedure are shown in table 4. The error rate of the procedure is 1.7%, so the error rate added to the whole process is 0.03%. It does not mean a significant drop in overall ambiguity, but it discards 40% of superfluous analyses.

[4] This group represents punctuation marks and other symbols.

	Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token	Precision
before	80.53%	4.23	3.60	92.31%
after	75.35%	2.98	2.49	90.42%

Table 4: Ambiguity measures on linguistic variants before and after the procedure

	Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token	Precision
initial	99.79%	18.06	18.01	98.12%
typographical	99.58%	8.18	8.15	96.46%
derivational	99.58%	7.94	7.91	96.46%
proper names	85.21%	6.93	6.05	95.94%
statistical 3+2+1	83.33%	3.99	3.49	91.98%

Table 5. Ambiguity measures on unknown words using all the procedures

	Distribution	Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token	Precision
standard	77.90%	80.73%	3.81	3.27	99.73%
variant	1.75%	75.35%	2.98	2.49	90.42%
unknown	2.65%	85.21%	4.06	3.61	93.02%
average	100.00%	66.46%	3.80	2.86	99.43%

Table 6. Ambiguity measures in the output of the improved analyser

However, this heuristic treats every rule equally, but not all of them have the same probability of being applied. We think that it could be interesting to use a probabilistic transducer (Mohri, 1997) to improve the precision measures of both the analyser and the disambiguation procedure of variants.

5.2. Disambiguation of unknown words

We have tested several procedures to detect and treat unknown words using different criteria:

1. Typographical disambiguation. Some analyses are discarded based on capital letters.
2. Disambiguation of derivational words to counterbalance overgeneration of the analyser. The goal of this procedure is to discard one of several interpretations when the morphological analyser assigns analyses as derivational and non-derivational word.
3. Identification and disambiguation of proper names not included in the lexicon. Some analyses can be disambiguated when identical lemmas are found in the same document.
4. Disambiguation based on both statistical and linguistic information. These statistics relates final trigrams of characters and POS tags. is used. The main features of the heuristic are: a) for each POS tag, leave at least one interpretation; b) assign a weight to each lemma according to the final trigram and the POS tag; c) select the lemma according to its length and weight –best combination of high weight and short lemma.

These procedures were designed to be applied consecutively. To decide the order in which they must be applied, we tried different combinations.

Finally, table 5 shows the best result of applying all the procedures in cascade.

This treatment has been designed to discard some of the interpretations of unknown words. Even if unknown words are only 2%-3% of the words, they constitute 15%-20% of the analyses. After applying the procedures, they only represent 3%-4.5% of the analyses, depending on the combination of procedures we use, and the average number of interpretations decreases from 18-19 down to 3,5-4,5. The overall results of treating the reference text are shown in table 8. This has been measured using the second level tagset both for disambiguation of linguistic variants and for statistical disambiguation of unknown words, thus leaving (at least) one lemma per class and subclass.

Precision decreases in average around 0.2%, even if the results for unknown words fall from 98% to 93%. Finally, we want to point out that each combination of the procedures may be used for different applications.

6. Improving precision

The main reason for these errors is the incremental architecture of the analyser. The first step in the process, the standard analyser, causes wrong interpretations, primarily when very short or very rare lemmas are involved in the analysis. However, the process stops when the analyser finds (at least) one interpretation of the word.

A clear example of these misinterpretations is *Barak*. This name, when it appears in its base form, is interpreted as *bara*, a common noun of very low frequency. When it appears inflected, i.e. *Barak-ek* (*Barak* in ergative case), the standard analyser assigns no interpretation and the analyser without lexicon interprets it correctly as a proper noun.

	Distribution	Ambiguity Rate	Interpretations per ambiguous token	Interpretations per token	Precision
standard	77.88%	81.02%	3.86	3.32	99.88%
variant	1.66%	81.36%	4.40	3.76	96.51%
unknown	2.76%	99.90%	18.20	18.18	98.34%
average	100.00%	67.21%	4.46	3.32	99.80%

Table 7: Ambiguity measures in the output of the analyser

Most of the errors are avoidable enriching the user's lexicon, but it is necessary to improve the results when this is not done.

So we must avoid rare and improbable analyses when a word has an initial capital letter. In order to avoid odd analyses we have marked short or conflicting lemmas with low probability as rare in the lexical database. Using this information, when all the possible interpretations for a word are marked as rare, the process follows using the next module. If at the next step the analyser does not find a non-rare analysis for the word, the word will be tagged just as the standard analyser did.

In the case of low frequency lemmas, words written with initial capital letter are also analysed by the guesser and only proper name interpretations are added to the ones suggested by the standard analyser.

In order to increase the precision in the analyser of linguistic variants, we limit the number of rules applied to obtain the interpretations. If all the interpretations have been obtained applying a higher value of rules than the threshold, the word will be treated using the guesser, thus, discarding the other interpretations.

We have implemented these proposals and the results are encouraging (see table 7). As a result, we have avoided 50% of the errors relaxing the constraints of the morphological analyser.

7. Conclusions

We have presented the work carried out to improve the robustness of a morphological analyser and to adapt it to new domains. We have made a proposal for the architecture of a morphological analyser combining different transducers to increase flexibility, coverage and precision. The design we propose is quite new as far as we know and we think that our design could be interesting for the robust treatment of other languages.

On the other hand, we have also defined some local disambiguation procedures, which don't take into account the context of the word, so as to discard many of the overgenerated analysis for non-standard words. The results of the research are very encouraging.

8. Acknowledgements

This work has been partial supported by the Education Department of the Government of the Basque Country **(**UE1999-2**)** and the Spanish Science and Technology Ministry (*Hermes* research project; 8/DG00141.226-14247/200).

We would like to thank to Xerox for letting us using their tools, and also to Ken Beesley and Lauri Karttunen for their help.

9. References

Aduriz I., I.Alegria, J. M. Arriola, X. Artola, A. Díaz de Ilarraza, N. Ezeiza, K. Gojenola and M. Maritxalar 1995. Different issues in the design of a lemmatizer/tagger for Basque. *From text to tag SIGDAT, EACL Workshop*.

Aizpurua I., I. Alegria, N. Ezeiza, 2000. GaIn: un buscador Internet/Intranet avanzado para textos en euskera. *Actas del XVI Congreso de la SEPLN*.

Aldezabal I., I. Alegria, O. Ansa, J. Arriola, N. Ezeiza, 1999. Designing spelling correctors for inflected languages using lexical transducers. *Proceedings of EACL'99*, 265-266. Bergen, Norway. 8-12.

Alegria I., M. Aranzabe, A. Ezeiza, N. Ezeiza, R. Urizar, 2001. Using Finite State Technology in Natural Language Processing of Basque. *6th Conf. on Implementation and Applications of Automata. CIAA'2001.*

Alegria I., X. Artola, K. Sarasola, M. Urkia, 1996. Automatic morphological analysis of Basque. *Literary & Linguistic Computing Vol. 11, No. 4*: 193-203. Oxford University Press.

Antworth E.L. 1990. *PC-KIMMO: A two-level processor for morphological analysis*. Occasional Publications in Academic Computing, No. 16, Dallas, Texas.

Arrieta B., X. Arregi, I. Alegria, 2001. An Assistant Tool For Verse-Making In Basque Based On Two-Level Morphology. *Literary and Linguistic Computing, Vol. 16, No. 1, 2001.* Oxford University press.

Black A., J. van de Plassche, B. Williams, 1991. Analysis of Unknown Words through Morphological Descomposition. *Proceedings of 5th Conference of the EACL*, pp. 101-106.

Ezeiza N., I. Aduriz, I. Alegria, J. M Arriola, R. Urizar, 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. *Proceedings of COLING-ACL'98*.

Karttunen L., 1993. Finite-State Lexicon Compiler. Xerox ISTL-NLTT-1993-04-02.

Karttunen L., 1994. Constructing Lexical Transducers, *Proceedings of COLING´94*, pp. 406-411.

Karttunen L., 2000. Applications of Finite-State Transducers in Natural Language Processing. *Proceedings of CIAA-2000.* Lecture Notes in Computer Science. Springer Verlag.

Karttunen L. and K. R. Beesley, 1992. *Two-Level Rule Compiler*. Technical Report Xerox ISTL-NLTT-1992-2.

Karttunen L., J.P. Chanod, G. Grenfenstette, A. Schiller, 1996. Regular Expressions for Language Engineering. *Natural Language Engineering, 2(4)*: 305:328.

Koskenniemi, K., 1983. *Two-level Morphology: A general Computational Model for Word-Form Recognition and Production*, University of Helsinki, Department of General Linguistics. Publications 11.

Mohri, M., 1997. Finite-state transducers in language and speech processing. *Computational Linguistics 23(2)*:269-322.

Oflazer K, C. Guzey, 1994. Spelling Correction in Agglutinative Languages. *Proceedings of ANLP-94*.

Oflazer K. 1996. Error-tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. *Computational Linguistics 22(1)*: 73-89.

Schiller A., 1996. Multilingual finite-state noun phrase extraction. In Wo*rkshop on Extended finite state models of language*, ECAI'96, Budapest, Hungary.

Sproat R., 1992. *Morphology and Computation*. The MIT Press.

Stede M., 1992. The Search of Robustness in Natural Language Understanding. *Artificial Intelligence Review 6*, 383-414.