Using Finite State Technology in Natural Language Processing of Basque

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

Ixa taldea. University of the Basque Country. i.alegria@si.ehu.es.

This paper describes the components used in the design and implementation of NLP tools for Basque. These components are based on finite state technology and are devoted to the morphological analysis of Basque, an agglutinative pre-Indo-European language. We think that our design can be interesting for the treatment of other languages. The main components developed are a general and robust morphological analyser/generator (Alegria et al., 96) and a spelling checker/corrector for Basque named Xuxen (Aldezabal et al., 99). The analyser is a basic tool for current and future work on NLP of Basque, such as the lemmatiser/tagger Euslem (Ezeiza et al., 98), an Intranet search engine (Aizpurua et al., 00) or an assistant for verse-making (Arrieta et al., 00).

1 Introduction

This paper describes the components used in the design and implementation of NLP tools for Basque. These components are based on finite state technology and are devoted to the morphological analysis of Basque, an agglutinative pre-Indo-European language. We think that our design can be interesting for the treatment of other languages.

The main components developed are a general and robust morphological analyser/generator (Alegria et al., 96) and a spelling checker/corrector for Basque named Xuxen (Aldezabal et al., 99). The analyser is a basic tool for current and future work on NLP of Basque, for example the lemmatiser/tagger *Euslem* (Ezeiza et al., 98), an Intranet search engine (Aizpurua et al., 00) or an assistant for verse-making (Arrieta et al., 00)

These tools are implemented using lexical transducers. A lexical transducer (Karttunen, 94) is a finite-state automaton that maps inflected surface forms to lexical forms, and can be seen as an evolution of two-level morphology (Koskenniemi, 83; Sproat, 92) where the use of diacritics and homographs can be avoided and the intersection and composition of transducers is possible. In addition, the process is very fast and the transducer for the whole morphological description can be compacted in less than one Mbyte. The tool used for the implementation is the *fst* library of *Inxight*[i] (Karttunen&Bessley, 92; Karttunen, 93; Karttunen et al., 96). Similar compilers have been developed by other groups (Mohri, 97; Daciuk et al., 98).

2 The design of the morphological analyser

The design that we propose was carried out because after testing different corpora of Basque the coverage was just about 95%. This poor result was due (al least partially) to the recent standardisation and the widespread dialectal use of Basque. In order to improve the coverage, we decided that it was necessary to manage non-standard uses and forms whose lemmas were not in the lexicon[ii], if we wanted to develop a comprehensive analyser. So three different ways were proposed: management of user's lexicon, analysis of linguistic variants and analysis without lexicon.

---

i Inxight Software, Inc., a Xerox Enterprise Company (www.inxight.com)

ii In some systems lemmas corresponding to unknown words are added to the lexicon in a previous step, but if we want to built a robust system this is not be acceptable.

We propose a multilevel method, which combines robustness and avoiding of overgeneration in order to build a general-purpose morphological analyser/generator. Robustness is basic in corpus-analysis but sometimes to obtain it overgeneration is produced. Overgeneration increases ambiguity and many times this ambiguity is not real and causes poor results (low precision) in applications based on morphology such as spelling correction, morphological generation and tagging.

The design we propose for robustness without overgeneration consists of three main modules (fig. 1):

1) The standard analyser using general and user's lexicons. This module is able to analyse/generate standard language word-forms. In our applications for Basque we defined —using a database— about 70,000 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). Because of 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 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 (names, 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 lemma-set with affixes related to open categories and general rules.

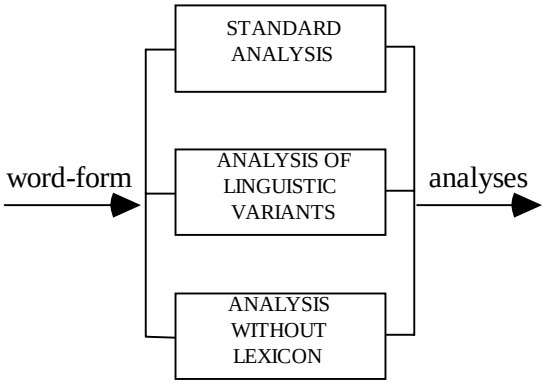

Figure 1.- Design of the analyser

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 can be seen as a variant of constraint relaxation techniques used in syntax (Stede, 92), 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. Relaxation techniques are used in morphology also by Oflazer (Oflazer, 96) but in a different way[iii].

With this design the obtained coverage is 100% and precision up to 99%.

[iii] He uses the term *Error-tolerant morphological analysis* and says: "*The analyzer first attempts to parse the input with t=0, and if it fails, relaxes t ...*"

The combination of three different levels of analysis and the design of the second and third levels are original as far as we know.

3 The transducers

3.1 Lexical transducers

A lexical transducer (Karttunen 94) is a finite-state automaton that maps inflected surface forms to lexical forms, and can be seen as an evolution of the two-level morphology where:

- Morphological categories are represented as part of the lexical form. Thus, diacritics may be avoided.
- Inflected forms of the same word are mapped to the same canonical dictionary form. This increases the distance between the lexical and surface forms. For instance *better* is expressed through its canonical form *good* (*good+COMP:better*).
- Intersection and composition of transducers is possible (Kaplan & Kay 94). In this way the integration of the lexicon, which will be another transducer, can be solved in the automaton and the changes between lexical and surface level can be expressed as a cascade of two-level rule systems where, after the intersection of the rules, the composition of the different levels is carried out (Fig. 2).

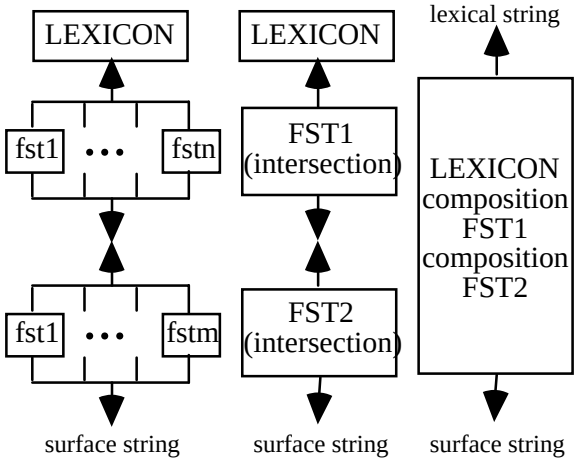

Figure 2.- Intersection and composition of transducers (from Karttunen et al. 92)

3.2 The standard transducer

Basque is an agglutinative language, that is, for the formation of words the dictionary entry independently takes each of the elements necessary for the different functions (syntactic case included). More specifically, the affixes corresponding to the determinant, number and declension case are taken in this order and independently of each other (deep morphological structure). One of the main characteristics of Basque is its declension system with numerous cases, which differentiates it from the languages spoken in the surrounding countries.

We have applied the two-level model but combining the following transducers:

1) **FST1** or Lexicon. Over 70,000 entries have been defined corresponding to lemmas and affixes, grouped into 170 sublexicons. Each entry of the lexicon has, in addition to the morphological information, its continuation class, which is made up of a group of sublexicons. Lexical entries, sublexicons and continuation classes all together define the morphotactics graph, i.e. the automaton

that describes the lexical level. The lexical level will be the result of the analysis and the source for the generation. This description is compiled and minimized in a transducer with 1.5 million states and 1.6 million arcs. The upper side of the transducer is the whole morphological information, and the lower side is composed of the morphemes and the minimal morphological information to control the application of the other transducers in cascade (FST2 and FST3).

2) **FST2**: Constraint of long-distance dependencies. Some dependencies among morphemes can be expressed with continuation classes because co-occurrence restrictions exist between morphemes that are physically separated in a word (Bessley, 98). For instance, in English, *en-*, *joy* and *-able* can be linked together (*enjoyable*), but it is not possible to link only *joy* and *–able (joyable*).* Using morphophonological rules is a simple way to solve them when, as in our system, it is only necessary to ban some combinations. Three rules have been written to solve long-distance dependencies of morphemes: one in order to control hyphened compounds, and two so as so avoid both prefixed and suffixed causal conjunctions (*bait-* and *–lako*) occurring together (*baitielako**). These rules have been put in a different rule system closer to the lexical level, without mixing morphotactics and morphophonology. The transducer is very small: 26 states and 161 arcs.

3) **FST3**: set of morphophonological rules. 24 two-level rules have been defined to express the morphological, phonological and orthographic changes between the lexical and the surface levels that happen when the morphemes are combined. Details about these rules can be consulted in (Alegria et al., 96) The transducer is not very big but it is quite complex. It is composed of 1,300 states and 19,000 arcs.

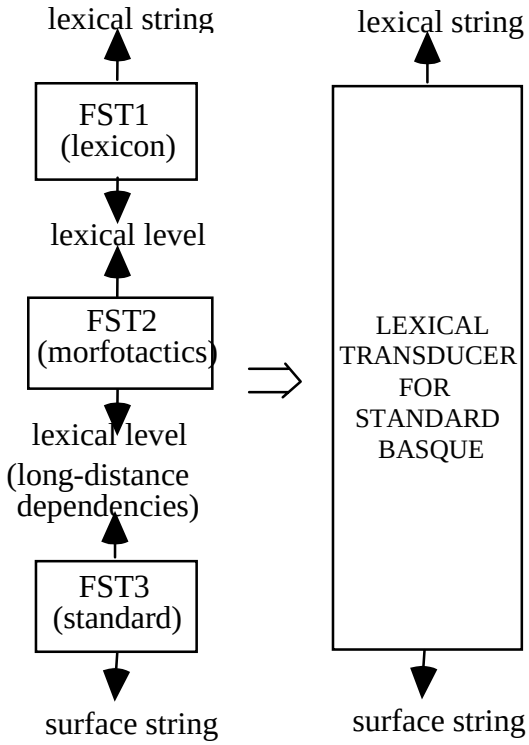

Figure 3.- Cascade of three transducers for standard analysis

The three transducers are combined by composition to build the standard analyser, which attaches to each input word-form all possible interpretations and its associated information. The composed transducer has 3.6 millions states and 3.8 million arcs, but is minimized into 1.9 M-states and 2 M-arcs, which take 3.2 Megabytes in disk.

A simple example of the language involved in the transducer is given in fig. 4

zuhaitz[zuhaitz][IZE_ARR]]+0[DEK_S_M]]+Etik[tik][DEK_ABL][iv]

FST1

zuhaitz++Etik

FST2

zuhaitz++Etik[v]

FST3

zuhaitzetik[vi]

Figure 4.- Example of cascade of transducer for standard analysis

3.3 The enhanced transducer

A second morphological subsystem, which analyses, normalizes, and generates linguistic variants, is added in order to increase the robustness of the morphological processor. This subsystem has three main components:

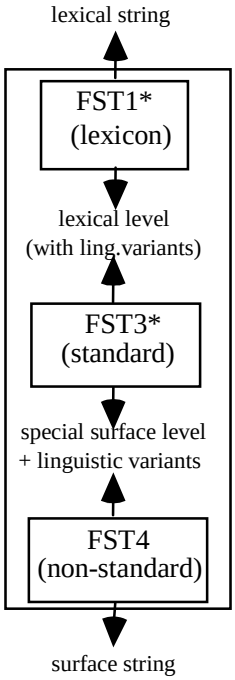

Figure 5.- Cascade of three transducers in the enhanced subsystem

---

[iv] *IZE_ARR*: common noun, *DEK_S_M* singular number, *Etik tik* suffix with epenthetical e, *DEK_ABL*: ablative declension case

[v] a rule in FST3 controls the realization of the epenthetical e (the next rule is a simplification): `E:e <=> Cons +: +: _` It can be read as "the epenthetical e is realized as e after a consonant in the previous morpheme"

[vi] *zuhaitzetik*: from the tree

1) **FST1***: New morphemes linked to their corresponding standard ones in order to normalize or correct the non-standard morphemes are added to the standard lexicon. Thus, using the new entry `tikan`, dialectal form of the ablative singular morpheme, linked to its corresponding right entry `tik` will be able to analyse and correct word-forms such `etxetikan`, `kaletikan`,... (variants of `etxetik` 'from the house', `kaletik` 'from the street', ...). More than 1500 additional morphemes have been included. Changes in the morphotactical information —continuation class— corresponding to some morphemes of the lexicon have been added too. In addition to this, the constraint of long-distance dependencies have been **eliminated** because sometimes these constraints are not followed, so FST2 is not applied. The compiled transducer for the enhanced lexicon increases the states from 1.5 to 1.6 millions and the arcs from 1.6 millions to 1.7..

2) **FST3*:** The standard morphophonological rule-system with a small change: the morpheme boundary (`+` character) is not eliminated in the lower level in order to use it to control changes in FST4. So, the language at this level correspond to the surface level enriched with the + character.

3) **FST4**: New rules describing the most likely regular changes that are produced in the linguistic variants. These rules have the same structure and management as the standard ones but all of them are optional. For instance, the rule `h:0 => V:V_V:V` describes that between vowels the `h` of the lexical level may disappear in the surface level. In this way the word-form `bear`, misspelling of `behar` (to need), can be analysed. As Fig. 5 shows, it is possible and clearer to put these non-standard rules in another level close to the surface, because most of the additional rules are due to phonetic changes and do not require morphological information. The additional rules do not need to be integrated with the standard ones, and so, it is not necessary to solve inconsistencies.

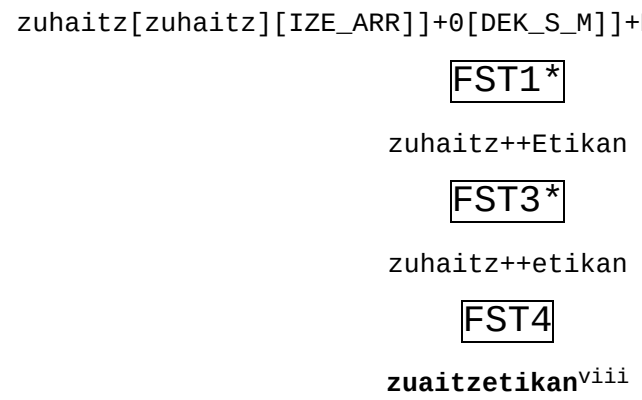

Figure 6.- Example of cascade of transducer for non-standard analysis

The composition of the FST1* and FST3* is similar in the number of states and arcs to the standard transducer, but when FST4 is added the number of states increases from 3.7 million states to 12 millions and the number of arcs from 3.9 millions to 13.1 millions. Nevertheless, it is minimized into 3.2 M-states and 3.7 M-arcs, which takes 5.9 Megabytes in disk.

3.4 The general transducer

The problem of unknown words does not disappear with the previous transducer. In order to deal with it, a general transducer has been designed to relax the need of lemmas in the lexicon. This transducer was

---

vii *IZE_ARR*: common noun, *DEK_S_M* declension singular number, *Etikan* dialectal variation of the *tik* suffix with epenthetical e, *DEK_ABL*: declension ablative case

viii *zuaitzetikan*: variation of *zuhaitzetik* (from the tree) with two changes: dropped h and dialectal use of *tikan*.

initially (Alegria et al., 97) based on the idea used in speech synthesis (Black et al., 91) but now it has been simplified. Daciuk (Daciuk, 00) proposes a similar way when he describes the *guessing automaton*, but the construction of the automaton is more complex.

The new transducer is the standard one modified in this way: the lexicon is reduced to affixes corresponding to open categories[ix] and generic lemmas for each open category, while standard rules remain. So, the standard rule-system (FST3) is composed of a mini-lexicon (FST0) 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. 7 the graph corresponding to the mini-lexicon (FST0) is shown.

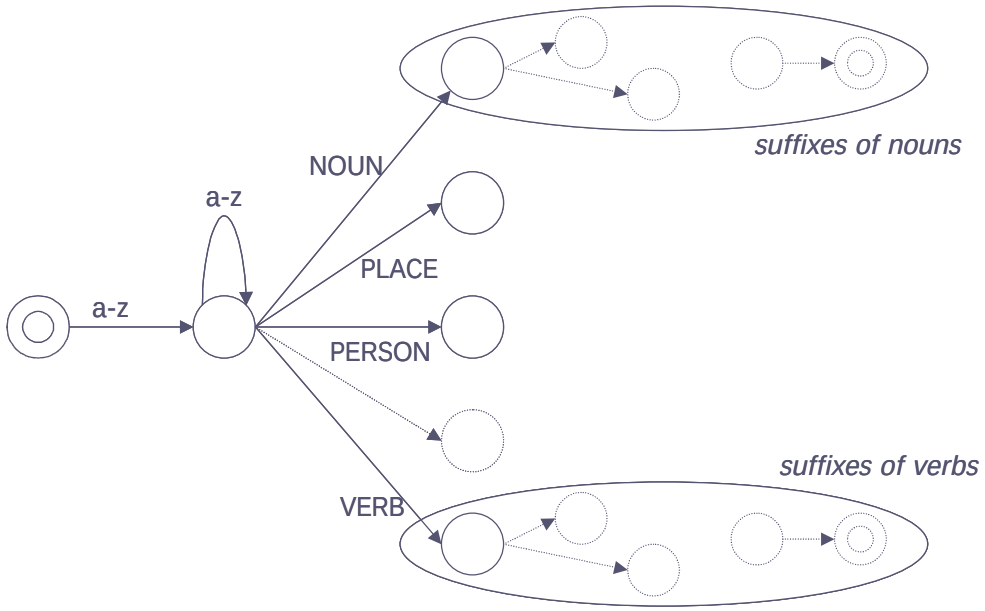

Figure 7.- Simplified graph of the mini-lexicon

The composed transducer is tiny, it is into 8,5 thousand states and 15 thousand arcs. Each analysis in the result is a possible lemma with the whole morphological information corresponding to the lemma and the affixes.

This transducer is used in two steps of the analysis: in the standard analysis and in the analysis without lexicon (named guessing in taggers).

In order to avoid the need of compiling the user's lexicon with the standard description, the general transducer is used in the standard analysis, and if the hypothetical lemma is found in the user's lexicon the analysis is added to the results obtained in the standard transducer.

If no results are obtained in the standard and enhanced steps the results of the general transducer will be the output of the general analyser.

3.5 Local disambiguation and ongoing work

Although one of the targets in the designed system is to avoid overgeneration, in the enhanced and general transducers overgeneration can still be too high for some applications.

[ix] there are seven open categories and the most important ones are: common nouns, personal names, place nouns, adjectives and verbs

Sometimes, the enhanced transducer returns analyses for words the lemmas of which are not included in the lexicon. That is to say, words that are not variants are analysed as such. Bearing in mind that the transducer is the result of the intersection of several rules each one corresponding to an optional change, the resulting transducer permits all the changes to be done in the same word. However, some combinations of changes seldom occur, so it is the general transducer that must accomplish the analysis.

Besides, sometimes there is more than one analysis as variant and it is necessary to choose among them. For example, analysing the word-form `kaletikan` (dialectal form) two possible analysis are obtained: `kale+tik` (from the street) and `kala+tik` (from the cove), but the first analysis is more probable because only one change has been done.

The solution could be to use a probabilistic transducer (Mohri, 97), or to improve the tool in order to obtain not only the lexical level but also the applied rules (this is not doable with the tools we have).

Currently, we use a local disambiguator that calculates the edit distance between the analysed word and each possible normalized word (generated using standard generation), choosing the most standard one(s) i.e. those with the lowest edit distance. Above a threshold, the results of this transducer are discarded. In the example above, `kaletikan` is compared to `kaletik` and `kalatik` (surface level of `kale+tik` and `kala+tik`). `kaletik` is chosen because its distance from `kaletikan` is shorter (2) than that of `kalatik`.

The general transducer presents two main problems:

- too many different tags can be produced. However, this problem is solved by a context based disambiguator (Ezeiza et al., 98)
- multiple lemmas for the same or similar morphological analysis. This is a problem when we want to built a lemmatizer. For example if `bitaminiko` (vitaminic) is not in the lexicon the results analysing `bitaminikoaren` (from the vitaminic) as adjective can be multiple: `bitamini+ko+aren`, `bitaminiko+aren` and `bitaminikoaren`, but the only right analysis is the second.

In the first case information about capital letters and periods is used to accept/discard some tags, but the second case is the main problem for us. A probabilistic transducer for the sublexicon with the set of letter-combinations would be a solution. However, for the time being, heuristics using statistics about final trigrams (of characters) in each category, cases, and lenght of lemmas are used to disambiguate the second case.

4 The spelling checker/corrector

The three transducers are also used in the spelling checker/corrector but, in order to reduce the use of memory, most of the morphological information is eliminated.

The spelling checker accepts as correct any word that allows a correct standard morphological analysis. So, if the standard transducer returns any analysis (the word is standard) or one of the possible lemmas returned by the general transducer is in the user's lexicon, the word is accepted.

Otherwise, a misspelling is assumed and the user gets a warning message and is given different options. One of most interesting option given is to include the lemma of the world in the user's lexicon. From then on, any inflected and derived form of this lemma will be accepted without recompiling the transducer.

For this purpose the system has an interface, in which the part of speech must be specified along with the lemma when adding a new entry to the user lexicon.

The proposals given for a misspelled word are divided in two groups: competence errors and typographical errors. Although there is wide bibliography about the correction problem (Kukich, 92), most

of the authors do not mention the relation between them and morphology. They assume that there is a whole dictionary of words or that the system works without lexical information. Oflazer and Guzey (1994) faced the problem of correcting words in agglutinative languages. Bowden and Kiraz (Bowden&Kiraz, 95) applied morphological rules in order to correct errors in nonconcatenative phenomena.

The need of managing competence errors —also named orthographic errors— has been mentioned and reasoned by different authors (van Berkel & de Smedt, 88) because this kind of errors are said to be more persistent and make a worse impression. When dealing with the correction of misspelled words the main problem faced was that, due to the recent standardisation and the widespread dialectal use of Basque, competence errors or linguistic variants were more likely and therefore their treatment became critical.

When a word-form is not accepted it is checked against the enhanced transducer. If the incorrect form is now recognised—i.e. it contains a competence error— the correct lexical level form is directly obtained and, as the transducers are bi-directional, the corrected surface form will be generated from the lexical form using the standard transducer.

For instance, in the example above, the word-form *beartzetikan* (misspelling of *behartzetik* "from the need") can be corrected although the edit distance is three. The complete process of correction would be the following:

- Decomposition into three morphemes: *behar* (using a rule to guess the h), *tze* and *tikan*.
- *tikan* is a non-standard use of *tik* and as, they are linked in the lexicon, this is the chosen option.
- The standard generation of *behar+tze+tik* obtains the correct word *behartzetik*.

The treatment of typographical errors is quite conventional and only uses the standard transducer to test hypothetical proposals. It performs the following steps:

- Generating hypothetical proposals to typographical errors using Damerau's classification.
- Spelling checking of proposals.

The results are very good in the case of competence errors —they could be even better if the non-standard lexicon was improved — and not so good for typographical errors. In the last case, only errors with an edit distance of one have been planned. It would be possible to generate and test all the possible words with a higher edit distance, but the number of proposals would be very big. We are planning to use the Oflazer and Guzey's proposal, which is based on flexible morphological decomposition.

4 Conclusions

In this paper we have presented an original methodology that allows combining different transducer to increase the coverage and precision of basic tools for NLP of Basque. The design of the enhanced and general transducers that we propose is new as far as we know. We think that our design could be interesting for the robust treatment of other languages

Acknowledgements

This work has had partial support from the Education Department of the Government of the Basque Country **(**reference UE1999-2**)**. We would like to thank Xerox for allowing us to use their tools, and also Lauri Karttunen for his help. Thanks to anonymous referees whose comments have helped us improving the paper.

References

Aizpurua I., Alegria I., Ezeiza N. (2000) GaIn: un buscador Internet/Intranet avanzado para textos en euskera. Actas del XVI Congreso de la SEPLN Universidad de Vigo, 26-28 septiembre de 2000.

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

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

Alegria I., Artola X., Ezeiza N., Gojenola K., Sarasola K. (1996) A trade-off between robustness and overgeneration in morphology. Natural Language Processing and Industrial Applications. Volume I. pp 6-10. Moncton, Canada. 1996.

Alegria I., Artola X., Sarasola K (1997). Improving a Robust Morphological Analyser using Lexical Transducers. Recent Advances in Natural Language Processing. *Current Issues in Linguistic Theory (CILT) series*. John Benjamins publisher company. Vol. 136. pp 97-110. 1997.

Arrieta B., Arregi X., Alegria I. (2000). An Assistant Tool For Verse-Making In Basque Based On Two-Level Morphology. Proceedings of ALLC/ACH 2000 . Glasgow, UK. 21-26 July 2000.

Bessley K. (1998). Constraining Separated Morphotactic Dependencies in Finite State Grammars. Proc. of the International Workshop on Finite State Methods in NLP. Ankara. 1998.

Black A., van de Plassche J., Williams B. (1991). Analysis of Unkown Words through Morphological Descomposition. Proc. of 5th Conference of the EACL, vol. 1, 101-106.1991.

Bowden T., Kiraz G. (1995). A morphographemic model for error correction in non-concatenative strings. Proc. of the 33rd Conference of the ACL, 24-30. 1995.

Daciuk J., Watson B., Watson R. (1998). Incremental Construction of Minimal Acyclic Finite State Automata and Transducers. Proc. of the International Workshop on Finite State Methods in NLP. Ankara. 1998.

Daciuk J. (2000). Finite State Tools for Natural Language Processing. Proceedings of the COLING 2000 workshop Using Toolsets and Architectures to Build NLP Systems, Luxembourg, 2000.

Ezeiza N., Aduriz I., Alegria I., Arriola J.M., Urizar R. (1998). Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. COLING-ACL'98, Montreal (Canada). August 10-14, 1998.

Kaplan R. M. and M. Kay (1994). Regular models of phonological rule systems. Computational Linguistics, vol.20(3): 331-380. 1994.

Karttunen L. and Beesley K.R. (1992). Two-Level Rule Compiler. Xerox ISTL-NLTT-1992-2.

Karttunen L., Kaplan R.M., Zaenen A. Two-level morphology with composition. Proc. of COLING´92. 1992.

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

Karttunen L. (1994). Constructing Lexical Transducers, Proc. of COLING´94, 406-411. 1994.

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

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

Koskenniemi, K. (1983). Two-level Morphology: A general Computational Model for Word-Form Recognition and Production, University of Helsinki, Department of General Linguistics. Publications nº 11, 1983.

Kukich K. (1992). Techniques for automatically correcting word in text. ACM Computing Surveys, vol.24, No. 4, 377-439

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

Oflazer K, Guzey C. (1994). Spelling Correction in Aglutinative Languages, Proc. of ANLP-94, Sttutgart. 1994.

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

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. 1992.

Van Barkel B, De Smedt K. (1988). Triphone analysis: a combined method for the correction of orthographic and typographical errors. Proceedings of the Second Conference ANLP (ACL), pp.77-83.