Eugenio Picchi
Istituto di Linguistica Computazionale, CNR, Pisa, Italy

Statistical Tools for Corpus Analysis: A Tagger and Lemmatizer for Italian

Abstract

We present the most recent addition to the PiSystem, an integrated set of tools for mono– and bilingual corpus creation and manipulation and dictionary construction. The new component is a statistical part–of–speech tagger and lemmatizer. The methodology adopted resembles that of similar procedures for other languages but the PiTagger has been developed to meet the particular requirements of a highly inflected language such as Italian. Texts analysed by the PiTagger can then be directly interrogated using the tagged corpus query procedures included in the system. The philosophy behind a procedure for sense disambiguation now being designed and tested is also briefly described.

1. Introduction

At Pisa, the last ten years have seen the development of an increasingly sophisticated set of tools known as the PiSystem. This system has been designed and implemented for mono– and bilingual corpus creation, management and querying, and includes a Lexicographic Workstation to handle all stages of dictionary construction. In the paper, the most recent addition to the system is described and our intentions for the future are presented. Descriptions of the other system components are given in Picchi (1991), Marinai et al. (1990, 1991).

The current trend in corpus linguistics is towards the construction of increasingly large text corpora. However, as the size of the corpora and the volumes of data to be processed grow, the need for a pre–processing of the incoming data so that the information can be filtered and extracted in more meaningful ways becomes more urgent. In particular, the users are demanding access to grammatically and semantically disambiguated corpora. The time and costs involved in the manual tagging of texts means that much attention is now being given to the development and implementation of methods for automatic text disambiguation. The PiSystem thus now includes a statistically–based procedure for the automatic lemmatization and PoS tagging of Italian texts. In addition, a procedure for sense disambiguation is now being designed and tested.

2. The PiTagger

The PiTagger has been designed to assign the grammatical category (PoS) and base lemma to all the word-forms in an Italian text. The tagger operates in two main steps: - morphological analysis of each word-form in the text under analysis and assignment of morph tags representing all the possible grammatical and lexical hypotheses; - automatic disambiguation of the morph tags using a statistical procedure based on a frequency value for each sequence of grammatical codes extracted from a Training Corpus and stored in a Reference Database. An interactive procedure, called TaggHand, can also be used, if necessary, for a rapid manual checking and correction of the results. Figure 1 shows the relationship between the different components forming the entire tagging and lemmatizing system. In the rest of this section, we will describe how these procedures operate.

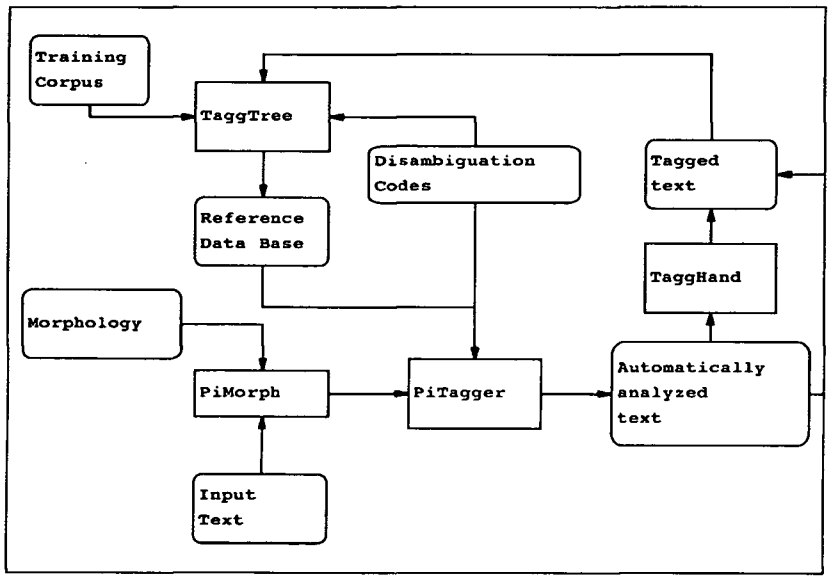

Figure1. PiTagger flow chart

2.1 Morphological analysis

In the first stage of the procedure, each word-form in the text being processed is input to a morphological analyser which identifies all its possible lexical and grammatical classifications, i.e. the lemmas and the grammatical categories to which the form (considered out of context) could belong. In addition to the grammatical category of the lemma, a classification of the form within its paradigm is given. All possible morphosyntactic information

is associated with each form of the original text. The module used by the procedure in this stage is the Italian component of the morphological engine implemented in the PiSystem. The lexicon file on which the rules that describe the Italian morphological system operate has been derived from the Italian Machine Dictionary (DMI). The results of this morphological analysis provide the input for the first stage of the automatic tagging process

```
95   Tali
96   conquiste
97   promettono
98 1,
99   per
100   il
101   futuro
102 1,
103   un
104   crescente
105 RR
106   impatto
107   sulla
108   vita
109   produttiva
110   e
111   sui
112   servizi
113   delle
114   società
115   industriali
116 RR
117   avanzate
118 1.
```

Figure 2 – Tokenized input text

```
95   2 TALI          TALE#PD@NP
                     TALE#DD@NP
                     TALE#A@NP
96   2 CONQUISTE     CONQUISTA#SF@FP
97    PROMETTONO     PROMETTERE#VTR@P3IP
98    ,
99    PER            PER#E@
100   IL             IL#R@MS
101   FUTURO         FUTURO#A@MS
                     FUTURO#SM@MS
102   ,
103   UN             UN#PI@MS
                     UN#R@MS
104   CRESCENTE      CRESCERE#VTI@NSPP
                     CRESCENTE#A@NS
                     CRESCENTE#SF@FS
105   RR
106   IMPATTO        IMPATTO#SM@MS
                     IMPATTARE#VIT@S1IP
107   SULLA          SULLA#E@FS
                     SULLA#SF@FS
108   VITA           VITA#SF@FS
109   PRODUTTIVA     PRODUTTIVO#A@FS
110   E              E#C@
111   SUI            SUO#PP@MP
                     SUI#E@
112   SERVIZI        SERVIZIO#SM@MP
113   DELLE          DELLE#E@FP
114   SOCIETA'       SOCIETA'#SF@FN
115   INDUSTRIALI    INDUSTRIALE#A@NP
                     INDUSTRIALE#SN@NP
116   RR
117   AVANZATE       AVANZARE#VITP@P2IP@P2MP
                     AVANZARE#VITP@FPPR
                     AVANZATA#SF@FP
                     AVANZATO#A@FP
118
```

Figure 3 – Output of morphological analysis

In Figure 2 we see an example of tokenized text which is fed into the morphological procedure. Figure 3 shows the output, in which the appropriate lemmas and morph codes have been assigned to each form. Obvously , the codes refer to an Italian analysis. Thus, in the case of the form 'CRESCENTE' the analysis gives three possible classifications:
CRESCERE – intransitive verb, present participle; CRESCENTE – adjective, singular; CESCENTE – feminine noun, singular.

2.2 Training corpus and reference database

In order to obtain the statistical data that is needed by the disambiguation procedure, a set of texts was manually lemmatised and tagged with morphosyntactic codes. A procedure then analysed the syntactical/ grammatical behaviour of the words in this Training Corpus (TC) and memorized the statistical results in a Reference Database. The choice of an optimal level of grammatical coding and therefore of number of codes used has been made on a trial and error basis and is still subject to adaptation. The particular morph codes employed, called Disambiguation Tags (DTs), have been defined in function of the system and are dependent on the solutions adopted and on the degree of distinction to be made. The initial intuitive decisions are now being refined on the basis of the first results.

In fact, the choice of the grammar codes used to tag the forms in the text is very important: the number of different codes employed must not be too low (for example, only the major POS's: V, N, A, etc.) as in this case the number of possible sequences of codes would be very small and most sequences would occur very frequently; vice versa, the use of a high number of codes, eg. tagging each word-form with a code that represents both its major part of speech and its particular inflection would produce a great number of sequences of codes, mostly with a very low frequency and therefore of little significance. Both of these extreme solutions present advantages and disadvantages: on the one hand, a low number of codes would be very easy to manage even with a Training Corpus of modest dimensions but would not supply sufficient information for the disambiguation algorithm; on the other hand, a large number of codes would lead to an excessive dispersion of information, an enormous growth in the volume of the data to be handled, and the need for a large Training Corpus in order to obtain sufficiently valid thresholds that would permit a correct evaluation of the frequencies of the different sequences of codes. It is thus clear that a satisfactory compromise between the two extremes must be found.

To give just one example, in the set of DT codes used for the verbs, the verbs *essere* (to be) and *avere* (to have) have been identified as special cases and tagged accordingly; for the other verbal classes particular subdivisions

have been introduced which are useful for our purposes: past participles, present participles, gerunds, and infinitive forms are coded separately, whereas one further code is used for the rest of the paradigm.

The rules that govern the conversion from the grammatical classification supplied by the morphological component and the DT code are held in an external file (Disambiguation Codes, see Figure 1) that can easily be modified to improve system performance. These disambiguation rules are needed by the PiTagger procedure both in the preparation of the Reference Data Base and also in the disambiguation of new texts.

Once the TC had been tagged, all possible sequences of three consecutive grammar codes (3–tuples) were calculated and, for each sequence recognised, its relative frequency in the corpus was computed. The results of the statistical analysis of the relative frequencies found in the Training Corpus for different sequences of DT's were then stored in the Reference Database. Each sequence of three DTs is assigned a numerical value that will be used by the disambiguation algorithm to calculate the probabilities for each possible sequence of 3 consecutive morphological tags found in the text being disambiguated.

The initial training corpus on which the procedure is being tested and evaluated is very small; it consists of a text of 50,000 words extracted from the Italian Reference Corpus. It is our intention to gradually extend this corpus by adding the first results (verified and corrected manually, if necessary) to it and then recomputing the relative frequencies for the Reference Database. In this way, the reference values used by the disambiguation procedure will be increasingly reliable and the success rate of the procedure should improve.

2.3 Disambiguation procedure

The text is processed in sequential order. Each word in the text, considered together with the tags that have been assigned to it by the morphological analyser, is examined 3 times, each time within a different set of 3 words (3–tuple), according to whether it is in first second or third position. Thus, for each 3–tuple, the different possible sequences of morph tags or DTs are calculated. For each word, a vector of 3 elements is then constructed, within which the procedure inserts the values extracted from the Reference Database relative to the likelihood of the word, in each of its three positions,

assuming one of its possible morph tags rather than another.

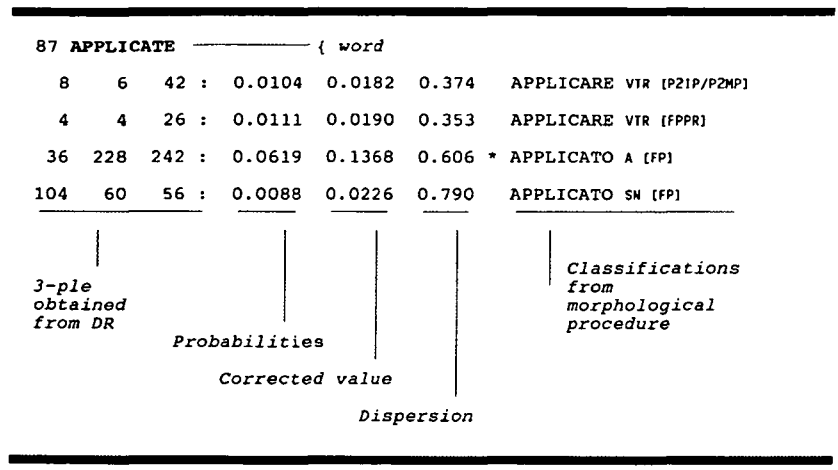

Figure 4 – Evaluated data in the disambiguation phase

A probability value is obtained by calculating the values of the vectors for each morph tag hypothesis for the current word; a dispersion value is then computed and a corrected value is calculated taking into consideration the dispersion value. Of all the solutions proposed for the current word, that to which the highest corrected value has been assigned is chosen as the right solution. Each time that a solution is chosen for a given word, all the values relative to the rejected solutions that had been inserted in the 3-ples of the words following the word under exam are automatically eliminated in order to prevent successive decisions being based on hypotheses that have not been accepted.

The concept of dispersion has been introduced in order to evaluate any relevant differences given by analyses made on the same word when this is considered within a triple in which it occupies three different positions (see Figure 5). In practice, constant values in a vector tend to confirm the hypotheses assumed, whereas very different values indicate that the hypothesis supplied by one triple is contradicted by those of the others. The dispersion attempts to measure such contradictions and uses them to correct the probability value. The dispersion measures three different values considering three different triples which refer to the same word.

The disambiguation system refers to an external table which lists pairs of adjacent grammatical codes for which a particular constraint on the agreement between person and gender should be imposed.

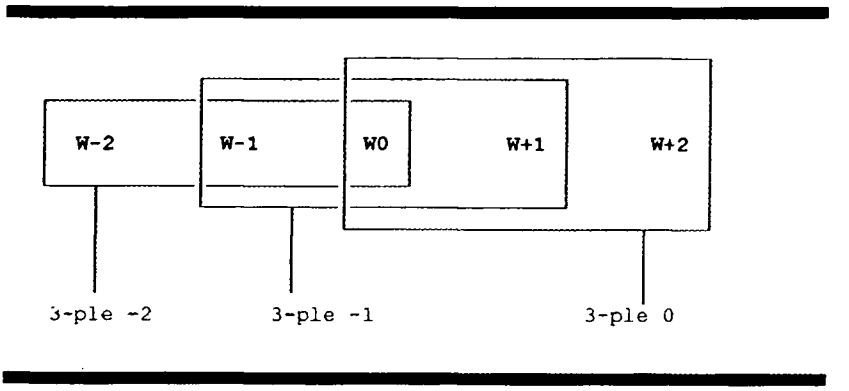

Figure 5. Word contextual environment

The algorithm disambiguates sequentially and operates on the computed values for each word in succession, with no backtracking in order to retest and verify solutions already chosen. At any given moment, the current word is the first word of the 3–uple being considered.

The reference dictionary used for Italian also contains information on usage. This information has been extracted and is assigned to the lemmas in order to prepare the way for an interesting extension to the disambiguation system. Each usage code has been transformed into a frequency value for the relevant lemma, and this value can be applied as a weight by the disambiguation algorithm. For example, when there is more than one possible solution, archaic words could be treated in two ways: (i) totally rejecting the solution tagged as archaic; (ii) considering the archaic solution to have the same value as the other possibilities, thus encouranging errors in analysis. However, we intend to apply a weight to the archaic possility, in order to influence appropriately the calculation of probability. This solution leaves the way open to other possibilities in the future, i.e. that of using sub–language dictionaries in which the words typical of that particular sector will be assigned a greater weight, or, once we have a large number of analyzed texts, creating a weighted dictionary, in which each entry has a relative frequency or probability assigned to it so that this weight can be applied in the disambiguation of new texts.

2.4 Tagghand

At the end of the automatic disambiguation procedure, the program will have assigned to each word-form in the text just one of the solutions originally proposed by the Morphological Component. Clearly, using a statistical procedure, a certain error rate must be expected and errors cannot be signalled automatically by the system. The error rate will depend on the size and degree of representability of the Training Corpus used, the type of text being analyzed, the correctness, appropriateness and detail of the Disambiguation Tags and the development, still under way, of particular checks to be added to the automatic procedure. In any case, a system of this type can never offer a solid 100 percent success rate. At present, the procedure described above gives an average success rate of 95%-97%, depending on the type of text being processed. This percentage is expected to improve as (i) the procedure is further refined on the basis of the first results, and (ii) more disambiguated texts are added to the Training Corpus and the values contained in the Reference Database are updated on the basis of a more consistent volume of statistical data.

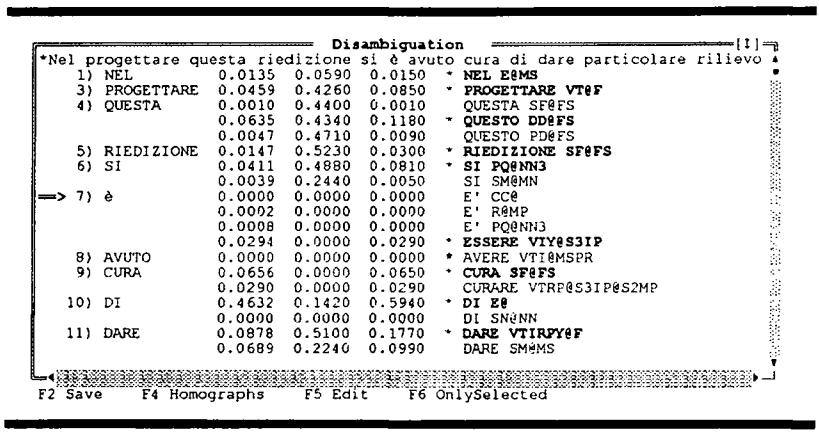

Figure 6

For many applications, the possibility of being able to grammatically tag and lemmatise rapidly and economically large quantities of texts may well be far more important than having a 100% correctness. However, for other applications, the total reliability of the results may be essential. For this reason, an interactive procedure, **Tagghand**, has been implemented in order to permit the user to scan the results of the automatic procedure quickly and, when necessary, intervene and correct them easily. An example of how the text is displayed on the screen to the user can be seen in Figure 6. The segment of text that has been analysed appears at the top of the screen. Each

word form is then listed with the various possible solutions, the one chosen by the systembeing marked by an asterix and a different colour. Using the mouse and the appropriate key functions it is very easy to correct wrong analyses. The key function "OnlySelected" permits the user to view the text with the only selected solutions.

Texts that have been PoS tagged and lemmatized using the PiTagger can then be input to the DBT management system which can also be used on tagged corpora. This system operates on the tagged data permitting the user to formulate queries which refer to the unannotated word–forms, to the word–form/tag couple, to the tags by themselves, e.g. particular sequences of tags can be searched, or to the lemmas in a text. It can thus be used as a flexible query system on tagged corpora, as one of the components of the Lexicographic Workstation for dictionary construction, or as a support in other kinds of applications. For full details see Monachini and Picchi (1992).

3. Next steps

While the PiTagger described above is in an advanced state of development and is already in use, a procedure for sense disambiguation is still in the design stage. Hence we will here give only an idea of the methodology currently being studied and tested. It is still too early to report any reliable results. Again, a statistically–based procedure and a Training Corpus are being used; in this case, in order to evaluate to what extent it is possible to automatically identify the senses of polysemous words in a text, on the basis of the relationship between the semantic class of the word under examination and that of the neighbouring words. At present, we are studying just two word classes: nouns and verbs.

The problem in text sense disambiguation is to select the correct sense of a polysemous item in a given context. The basic assumption underlying our procedure is that a term used in a particular sense will normally be found in similar, recognizable "contexts", i.e. in cooccurrence with similar groups of words. If the sense of a word changes, the "context" will also be different. The "context" of a word is here determined by the semantic codes assigned to the neighbouring words.

In the PoS tagger and lemmatizer described in the previous section, the codes used for tagging were defined as the result of a manual analysis of a training corpus, and the relative probabilities of different sequences of codes were then calculated. We are following a similar methodology here. The system is based on the concept of Disambiguation Tags (DTs) – in this case referring to semantic rather than syntactic data. In the TC for the sense disambiguation procedure, each word (nouns and verbs only at present) is assigned a code which defines its particular semantic class. The possible

semantic codes for these words are extracted from an electronic Reference Dictionary (2). In this dictionary, a DT denoting its semantic code is assigned to each sense of every noun and verb headword. In the case of nouns, the DT used is a term identifying the superordinate or genus term in the definition. This taxonomic information has already, to a large extent, been stored in the dictionary as a result of the definition parsing procedures described in Hagman (1992). The data is now being revised manually to ensure that the classification is significant for the purposes of the sense disambiguation algorithm. For verbs, the assignment of the DT is more complex. As it is not always possible to assign verbs to a taxonomy, we have decided to associate each sense of each verb with what we call a "quasi–synonym" class. The code assigned manually to this class is used to tag the verbs of this group semantically. In the same way as for the morph codes in the PiTagger, the level of semantic coding is crucial. If too generic terms are selected, then we will have an insignificant number of tags and code sequences, and the procedure will be unable to choose correctly between possible solutions.

A Reference Database is being generated from the Training Corpus on the basis of the DTs that have been assigned to the word–forms (nouns and verbs) in the Training Corpus. This Reference Database will provide the statistical data to be used by the algorithm when disambiguating a text to determine the most likely sense for each polysemous term. The procedure is based on an analysis of the co–presence of pairs of elements (i.e. DTs) in a context window and the selection of the most probable tags within the given context. The criteria to be used to cut this context window and sets its dimensions are now being defined.

Notes

1 The DMI contains approximately 120,000 Italian lemmas together with the information necessary to recognize and generate all the associated forms.
2 Our Reference Dictionary is based on the Garzanti Italian dictionary, one of the lexical components of the PiSystem.

References

Hagman, J. 1992. "Semantic Parsing of Italian Dictionary Definitions", ACQUILEX Working Paper No. 047. Pisa: Istituto di Linguistica Computazionale.

Monachini M. and Picchi E 1992. "Tagged Corpora: A Query System" in F. Kiefer, G. Kiss, J. Paijzs (eds.), *COMPLEX '92, Papers in Computational Lexicography*, 229–236. Budapest.

Marinai, E., Peters, C. and Picchi, E. 1990. "The Pisa Multilingual Lexical Database System", Esprit BRA 3030, Twelve Month Deliverable. Pisa: Istituto di Linguistica Computazionale.

Marinai, E., Peters, C. and Picchi, E. 1991. "Bilingual Reference Corpora: A System for Parallel Text Retrieval" in *Using Corpora, Proc. of 7th Annual Conference of the UW Centre for the New OED and Text Research*, 63–70. Oxford: Oxford University Press.

Picchi, E. 1991. "D.B.T.: A Textual Data Base System", in L. Cignoni and C. Peters (eds.), *Computational Lexicology and Lexicography. Special Issue dedicated to Bernard Quemada. II, Linguistica Computazionale*, VII, 177–205. Pisa: Giardini Editore.