VEST – TAGGER SIMBOLICO DI VENEZIA

VEST - VENICE SYMBOLIC TAGGER

RODOLFO DELMONTE

SOMMARIO/ ABSTRACT

In questo articolo descriviamo il tagger per l'italiano chiamato VEST che utilizza risorse lessicali e una tabella di trigrammi per derivare i valori frequenziali di triplette di tag. Abbiamo utilizzato un tagset di 75 etichette basato su un nostro precedente schema di tagging per l'italiano e quindi abbiamo trasformato quei tag che lo richiedevano nel tag appropriato o più vicino al tagset proposto nella sfida EVALITA. I risultati sono più bassi della media di base e richiedono ulteriore lavoro nell'individuare le eccezioni e nell'affinare i parametri.

In this paper, we describe the tagger for Italian called VEST which uses lexical resources and a trigram table to derive frequency values of tag-triples. We used a tagset of 75 tags based on our previous tagging scheme for Italian and then trasformed those tags that required it into the appropriate or closest tag to the tagsets proposed in the challenge EVALITA. In a number of cases we had to provide additional information and make subcategorization frames available. In other cases specific lexical classes had to be made available. Results are lower than average baseline and require more work in sorting out exceptions and tuning parameters.

Keywords: Tagging, Lexica, Symbolic Rules.

1. Introduction

Vest is a symbolic rule tagger that uses little quantitative and statistical information. Most of its computational work is based on tagged lexical information available in datasets made available from previous work in the field. The system also uses a morphological analyzer which is only activated for derivational nouns, cliticized verbs and some adjectives. It is also activated as a guesser by unknown, and out of vocabulary words which will end up with a default classification in case of failure: uppercase words are labeled proper nouns, lowercase words common nouns. The guesser looks at first for a legal morphological decomposition.

As to quantitative information it is organized in the following three databases:

- IWL – the Italian Word List – contains some 30K wordforms accompanied by tag/s and overall frequency count in a corpus of 1 million tokens; and Freqss another list of 28K wordforms with frequency of occurence in a 300K corpus.
- TrigramsIt which contains some 7000 trigrams of tags with their right and left context, which we use to assign cooccurence probabilities to ambiguous strings.

As to the remaining lexical resources and their size in terms of items listed, we report the whole set here below.

This is the list of all lexical resources made available to the tagger with their contents.

MIDUV - Dictionary of Italian 72000 lemmata**;** VITSTYPES - 22500 wordforms types with tags; ALLPOLS - 9800 Polywords tagged; LEXIT - 12200 unsuual wordforms including names; ROOTIT - 60000 roots for morphological analysis fully classified; ITALDICT - 12000 all upper case names and geographical nouns ; LIFUV - 17000 fully subcategorized verb entries; ITALLEM - 33000 lemmatized verb wordforms derived from Pisa online lexical material; FREQS - VIT Types with frequencies and tags; TRIMDEVS - trigrams derived from Development Set; ALLDEVS - 21500 word forms with tag derived from Development Set.

2. The Algorithm

The algorithm is organized as follows:

1. ***read_tfile***(X, Phrase)
2. ***ttagtext***(Frase,Outs)
3. ***disambs***(Outs,Disoutss)
    a. ***tevaluate***(Disoutss,Disouts,Output),
    b. ***memouts***(Disouts),
4. ***attempteagl***(Output,Catss),

1. ***read_tfile*** reads one sentence at a time - based on punctuation only - and passes it to the tagger;
2. in ***ttagtext*** the tagger recursively tries to assign one or more tags to each word of the entry sentence - this is done by means of the 75 tags tagset;
3. then with ***disambs*** the disambiguation phase takes place, and starts by looking up the list of available wordform types tagged according to the tagset under

experiment: in this list of types each wordform can be accompanied by a unique or an ambiguous list of tags which is associated to the previous choice. Then the disambiguator chooses the best and less costly choice: this is done easily whenever an unambiguous choice is available. Whenever two adjacent words are ambiguous both in the original tagset and the current one the disambiguator will look for frequency counts for trigrams and will also collect frequency counts associated to each wordform for that tag, if available or in IWL. Probabilities are produced by dividing the total frequency count for a given wordform by the frequency count associated to the current trigram.The disambiguator will try to resolve the ambiguity as soon as possible but basically as soon as a non ambiguous word arrives in the three words window. Probabilities are multiplied locally by the following trigram associated to the new window configuration.

4. Finally in the last call ***attempteagl*** the Output is transformed into the Eagle Tagset.

The recursive call to the lexical resources is organized to check the input word for the following information and tags accordingly:

- it is an integer and tags as "num"; it is in the list of unique tag words; it is a punctuation mark; it is a multiwords; it is in an uppercase list of location words; it is in an uppercase list of temporal words; it is in the lemmata dictionary; it is in the verb lemmata dictionary; decides with heuristics whether it is an abbreviation; it is a special number or a formula;
- if none obtains it activates the morphological analyser
- the it activates the guesser
- eventually, it assigns "nw" as label for unknown words

Before entering the actual disambiguation phase, the disambiguator builds its bivalent structure where tags belonging to the extended tagset are accompanied by tags belonging to the Development set. This is done in a first run. Statistical disambiguation is then preceded by a pass through all exceptions, words or classes of words which are highly ambiguous and would not be properly tagged solely on a quantitative basis. Some such words are "stato-stati", "che", etc.

Finally, the statistical disambiguation looks through the input string in search of ambiguously tagged words and tries to solve the ambiguity as soon as possible. In some cases this takes place locally, within the three words windows, in case the ambiguous words is followed by a non ambiguous word. Results and Future Work Results are not very satisfactory. We did some additional work in the meantime and more work will be done before the workshop to complete the exception list and to experiment with different parameters and settings.

Tab. 1 Results for 1st and 2nd run on both tagsets

1st RUN	
DISTRIB	
GLOBAL DATA: 1486 differences on 17313 tokens	
Accuracy = 91.42	**Error Rate = 8.58**
UNKNOWN TOKENS: 175 differences on 1326 tokens	
UTAccuracy = 86.80	**UTError Rate = 13.20**
EAGLES	
GLOBAL DATA: 1411 differences on 17313 tokens	
Accuracy = 91.85	**Error Rate = 8.15**
UNKNOWN TOKENS: 206 differences on 1326 tokens	
UTAccuracy = 84.46	**UTError Rate = 15.54**
2nd RUN only EAGLES	
GLOBAL DATA: 1154 differences on 17313 tokens	
Accuracy = 93.33	**Error Rate = 6.67**
UNKNOWN TOKENS: 174 differences on 1326 tokens	
UTAccuracy = 86.88	**UTError Rate = 13.12**

REFERENCES

[1] Delmonte R., G.A.Mian, G.Tisato. Un riconoscitore morfologico a transizioni aumentate, Atti Convegno Annuale A.I.C.A., Firenze, pp. 100-107, 1985.

[2] Delmonte R., E.Pianta. IMMORTALE - Analizzatore Morfologico, Tagger e Lemmatizzatore per l'Italiano, in Atti Convegno Nazionale AI*IA Cibernetica e Machine Learning, Napoli, pp. 19-22, 1996.

[3] Delmonte R, Luminita Chiran, Ciprian Bacalu. Elementary Trees For Syntactic And Statistical Disambiguation, Proc.TAG+5, Paris, pp. 237-240, 2000.

CONTACT

RODOLFO DELMONTE
Dipartimento di Scienze del Linguaggio, Università Ca' Foscari di Venezia
Email: delmont@unive.it

RODOLFO DELMONTE è Associato in Linguistica ed è responsabile del curricolo in Linguistica Computazionale del Corso di Laurea in Scienze del Linguaggio presso l'Università Ca' Foscari di Venezia. I suoi interessi vanno dallo Speech al Text Generation, dalla prosodia agli strumenti di CALL, dalla Sintassi LFG per il parsing alla Situation Semantics per il NL Understanding. Recentemente ha lavorato a progetti di annotazione linguistica per un Treebank dell'italiano scritto e parlato; inoltre ha creato un sistema per il Text Entailment in inglese completamente simbolico.