Accurate Annotation: An Efficiency Metric

ANTÓNIO BRANCO & JOÃO SILVA
University of Lisbon

Abstract

When accurately tagging corpora for training and testing automatic POS taggers, research issues tend to focus on how to ensure the integrity and consistency of the data produced. The issues we address here are instead: Provided that integrity and consistency are guaranteed, how to evaluate the complexity and efficiency of different accurate tagging procedures?

1 Introduction

While for automatic tagging, both issues of efficiency and accuracy have been addressed, for hand tagging, in turn, research tends to focus around the issue of accuracy (Voutilainen 1999). This is understandable as it is full accuracy that is sought by the hand tagging as this is its specific responsibility in the production cycle of automatic taggers. However, efficiency is definitely also a relevant issue here: provided integrity and consistency of the data can be ensured, what is the most efficient way — requiring the least amount of time — to annotate a corpus with full accuracy?

This question is not only relevant in itself but also because there is a growing perception that the way to obtain better automatic taggers is to have larger training corpora than the ones that have been used so far. Results by Banko & Brill (2001) suggest that the learning curve for machine learning techniques used to extract linguistic knowledge of various kinds, including for POS tagging, is log-linear at least up to 1 billion tokens of training size. Moreover, for the specific task studied in that paper, the best performing algorithm when the training size is 1 million tokens turns out to be the worst performing one when the training size is increased by 3 orders of magnitude. These authors underline that this "suggests that it may make sense for the field to concentrate considerably more effort into enlarging our training corpora and addressing scalability issues, rather than continuing to explore different learning methods applied to the relatively small extent training corpora" (p.1).

As the motto becomes "the larger the better", even if it will be possible to design better techniques to compress training data, the need for massive hand labelling not only does not disappear but increases given there seems to exist no limit for the desirable increase of the size of accurately labelled corpora and consequently for the increase of annotation labor.

The question about the efficiency of accurate tagging is thus relevant. To seek an answer for it, one needs a sensible methodology to objectively estimate

the complexity of the manual tagging of a corpus; and to evaluate and rank major procedures followed for accurately tagging corpora according to their complexity.

2 An objective metric for annotation efficiency

An objective metric for tagging complexity has to be designed in to allow to determine: With respect to a given corpus and two different annotation procedures, which one is the most efficient; And with respect to a given annotation strategy and two corpora (possibly at different stages of development), which corpus implies the largest annotation complexity.

Such a metric has to measure what is essential for the complexity of the annotation task and abstract away from contingent aspects. It has also to be predictive in the sense that it should not need full or partial completion of an annotating task for the complexity of this task to be known beforehand.

Asking a human, or a group of humans, to tag a portion of a corpus and then registering the amount of time consumed by them will not do. Not only this fails to support a predictive viewpoint, as it does not abstract away from nonessential aspects: Differently skilled annotators will take different amounts of time to complete a task of the same size; different ergonomic environments will induce different working rhythms, etc.

Abstracting away from contingent aspects, what remains as the essential dimension in the complexity of tagging is the number of decision steps required. The number of decision steps to annotate a corpus is proportional to the number of tokens that exist in that corpus to be annotated. It is also proportional to the number of tags in the tag set, that is to the number of tags that have to be excluded to determine the tag to be assigned (i.e., the size of the search space to find the correct tag).

If stripped away from contingent details, hand annotation can be conceptualised as the procedure performed by an agent using the following facilitating tool: When in a given point of a corpus, the tool jumps to the next token to be tagged, skipping over the tokens that need not be tagged; after a token is selected, the agent scans through a list of tags (e.g., in a drop down menu) and choose which one to assign. To abstract from possible differences of the agents and isolate the intrinsic complexity of the procedure, what is measured is not the time consumed but the number of steps necessary to complete the assignment task.

Accordingly, to determine the complexity of hand tagging from scratch a corpus, one just needs to know the size of the corpus and of the tag set. If the corpus has N tokens and the tag set T tags, for each token, $(T + 1)/2$ decision steps are needed on average to find the tag to be assigned.[1] Hence, the complexity

[1] Under the assumption that the tags are equiprobable: More on this below.

of accurately tagging that corpus is proportional to $N[(T+1)/2]$ of what we termed standard tagging steps (sts).

Two important comments are in order here before proceeding. First, this metric is not aimed at providing a measure to determine the complexity involved in annotating a specific token in a specific occurrence: There is no pretension that the abstract procedure described above has any kind of psychological reality. Second, this metric is not aimed either at determining an absolute measure of the complexity of annotating a specific corpus: There is no pretension that the best device to support the hand annotation of a corpus should be one that permits picking one tag in a drop down menu, as a simple voice command might well be envisaged as a better alternative. As we hope to make clear in the discussion below, this metric is to be used as an objective basis to establish, in general terms, a comparative assessment of the average efficiency of different methods that might be conceived to accurately hand annotate a corpus with POS tags.

3 Accurate tagging

Turning now to the identification of the major conceivable procedures to annotate a corpus, we will be assuming that the preparatory preliminaries had been accomplished, the tag set was defined, the facilitation tool is in place, etc.

3.1 *From-scratch method*. The baseline strategy to accurately tag a corpus consists simply in hand annotating it token by token with the appropriate tags.

3.2 *Train-tag-review method*. On a par with this strategy, a method has been advocated based on a "bootstrapping" approach. As far as we were able to trace, its first proposition in a published paper was due to (Day et al. 1997).

This procedure consists in hand tagging a portion of the corpus and using the outcome as seed data to train a first tagger. This tagger is used to tag another stretch of text. The annotation produced is reviewed. The two portions of text are put together and used to train a second tagger, with better accuracy than the first. This cycle can be iterated, with increasingly larger portions accurately tagged and increasingly accurate automatic taggers.

3.3 *Sieve method*. Another "bootstrapping" procedure of a different kind, based in a two-step strategy, can yet be conceived. In the second step, the human annotator just tags the portion of the corpus whose tagging was not terminated in the first step. The first step is accomplished by a "sieving" tool. This tool reduces both the number of tokens that are left to be tagged and the size of the list of admissible tags to be assigned by the human annotator in the second step.

This procedure explores regularities concerning so-called closed and open classes of words.

Closed classes: The few hundreds lexical items of closed classes exhibit very high frequencies. With the help of library reference grammars and online dictionaries: (i) collect the list of those items not pertaining to the classes of Common

Nouns, Adjectives, Verbs, Adverbs ending in *-ly*, Proper Names;[2] (ii) associate each such item with the exhaustive list of its admissible tags (either from closed or open classes); (iii) assemble a sieving tool that by simple lexical lookup, when run over a corpus, tags the tokens of each type collected in (i) with their admissible tag(s).

Open classes: Many items from open classes result from a few derivational processes. As a rule, these processes and the categories of the resulting words can be identified from the word endings. With the help of reference grammars and online dictionaries: (i) collect a list of those word endings and corresponding categories; (ii) for each word ending, collect the corresponding exceptions, i.e., words with that ending but with a category not resulting from the corresponding morphological process;[3] (iii) extend the sieving tool so that after it has tagged closed class tokens, it detects tokens from open classes that bear endings like those in (i) or are one of the exceptions in (ii), and assigns them the admissible tag(s).[4]

This sieving/tagging tool designed along these lines performs the first, sieving step. In the second step, tokens tagged with only one tag are skipped by the human anotator as they are already accurately tagged. To annotate tokens that receive more than one tag, it is enough to pick one of the tags previously assigned. For tagging tokens with no tag yet, it is enough to pick one of the few tags of the open classes left: As adverbs ending in *-ly* were dealt with by the sieving tool, there will be four tags to opt for — Common Nouns, Adjectives, Verbs and Proper Names.[5]

4 Efficiency

In order to check the effectiveness of the metric for tagging complexity, it is now used over each tagging procedure just described. For the sake of having a concrete basis for discussion, let us assume that our task was to accurately tag a corpus with, say, 1 million tokens with an average sized tag set with 39 tags (cf. Annex). In what follows, it will become apparent that the comparative results arrived at would not be affected in case the discussion example opted for was different.

[2] See the tag set used in the Annex

[3] E.g., `ally_CN` is an exception to the rule that assigns `ADV` to tokens ending in *-ly*.

[4] This tool is easily extended to tag also numbers and other tokens that can be described by regular expressions.

[5] Full accuracy is ensured by the fact that if an item was tagged in the first step, it is because it is in the list used by the sieving tool; if it is entered into this list, it is possible and easy to make an exhaustive collection of all its admissible categories. Full coverage of closed classes (or derivationally obtained words from open classes), in turn, is not easy to ensure, but this is harmless: if an item is not entered in the list used by the sieving tool, that item does not receive any tag in the first step and will be found (and accurately tagged) in the second step by the human annotator.

4.1 *From-scratch*

Upper bound. With the from-scratch procedure, we will need to decide which tag to choose for each of the 1 million tokens out of a tag set with 39 tags. If the tags had identical probability of being selected, the annotation of our working corpus would require 20 Msts = $10^6 \cdot [(39+1)/2]$.

Lower bound. However, the tags are not equiprobably selected as different classes of words have different frequencies. The above value for complexity can be reduced if the list of tags presented to the annotator in the drop down menu of the facilitating tool (from which he selects the one to assign) is ranked by the decreasing frequencies of the tags. Hence, a more frequent tag will need less decision steps to be assigned than those required by a less frequent tag (cf. Annex).

To recalculate the complexity value, we take the typical values for the relative frequencies of different classes. This permits to rank the tags and determine how many steps each tag requires to be assigned. In this paper, we use the frequencies of the tags in the Annex observed in an accurately tagged corpus of Portuguese.[6]

The task of annotating, e.g., the Adverbs in our working corpus, will now require $7 \times 0.0529 \times 10^6$ sts as the assignment of tag ADV to each of the 0.0529×10^6 adverbs takes seven steps.

The lower bound for the complexity of our tagging task is obtained by the summation of the similarly computed values for every tag. This amounts to 5 194 129 sts.[7] The complexity of the from-scratch procedure is thus in the range 5.2–20.0 Msts.

4.2 *Train-tag-review*

To estimate the complexity of the train-tag-review procedure, there is a range of options concerning the size of the portion to be used as seed data and the size of each portion to be covered in subsequent tag-review iterations.

Upper bound. *Step 1:* Considering the learning curves in (Brants 2000), for the sake of simplicity, we assume that an initial, 90% accuracy tagger can be obtained with a training corpus of 10 Ktokens. Annotating a text of this size with the facilitating tool with the tags ranked by decreasing frequencies has a complexity of 51 941 sts.[8]

Step 2: To improve the accuracy of the tagger from 90% to 95%, the training corpus is enlarged from 10 to 100 Ktokens. This is done by running the initial tagger over a new portion of 90 Ktokens and then reviewing the outcome. 81

[6] For a discussion of this option, see Section 6. The corpus used has 260 Ktokens accurately tagged with the tag set in the Annex. It contains excerpts from news articles, magazines, and fiction novels. We are grateful to Fernanda Nascimento and Amália Mendes (CLUL) for having granted access to it.

[7] $= 10^6 \times (1 \times 0.1537 + 2 \times 0.1445 + 3 \times 0.1439 + \cdots) = 5\,194\,129.$

[8] $= 10^4 \times (1 \times 0.1537 + 2 \times 0.1445 + 3 \times 0.1439 + \cdots) = 51\,941.$

Ktokens (90% of 90) will be correctly and 9 Ktokens incorrectly tagged. In the reviewing process, each of the 81 Ktokens needs 1 sts to confirm its tag, and each of the remaining 9 Ktokens need as many steps as if it was to be annotated from scratch. For the latter, 46 747 sts will be needed.[9] The annotation of this second portion of 90 Ktokens requires thus 136 747 sts.[10]

Step 3: Let us assume that 97% accuracy can be reached with a corpus of 1 Mtokens. This can be achieved by running the last tagger over a new portion of 900 Ktokens and then reviewing the result. 855 Ktokens (95% of 900) will be correctly and 45K incorrectly tagged. As in step 2, each of the 855 Ktokens requires 1 sts to confirm its tag; each of the 45 Ktokens needs as many steps as if it was tagged from scratch: 233 736 sts. The annotation of this third portion with 900K thus requires 1 088 736 sts.[11]

The task of tagging the working corpus is now complete and the value for its complexity is obtained by summing up the values obtained in each of the above steps. The result is 1 272 230 sts.[12]

Lower bound. A lower bound for the complexity of the tag-train-review procedure is obtained by assuming that we start already with a 98% accuracy tagger, previously developed upon independent training data. This means that the training steps will be bypassed.

After running this tagger over the 1 million corpus, the tags assigned to 98% of it need to be confirmed, while the other 2% are to be reviewed. Each of the 980 Ktokens (98% of 1M) requires 1 sts to confirm its tag. Each of the remaining 20 Ktokens requires as many steps as if it was to be tagged from scratch. This involves 103 883 sts.[13] Taking these values together, the lower bound is determined as 1 083 883 sts.[14]

Considering the figures obtained for the upper and lower bounds concerning the tag-train-review procedure, its complexity is estimated as lying in the range 1.1–1.3 Msts.

4.3 *Sieve*

To estimate the complexity of the new sieve method, we implemented a sieving tool along the lines described in the Section 3.3.[15] A lower bound for the complexity of this method is calculated assuming that this sieving tool has a heuristics to detect Proper Names with 100% precision and recall (on the basis

[9] $= 9 \times 10^3 \times (1 \times 0.1537 + 2 \times 0.1445 + 3 \times 0.1439 + \cdots) = 46\ 747.$

[10] $= 90\ 000 + 46\ 747 = 136\ 747.$

[11] $= 855\ 000 + 233\ 736 = 1\ 088\ 736.$

[12] $= 46\ 747 + 136\ 747 + 1\ 088\ 736 = 1\ 272\ 230.$

[13] $= 20 \times 10^3 \times (1 \times 0.1537 + 2 \times 0.1445 + \cdots) = 103\ 883.$

[14] $= 980\ 000 + 103\ 883 = 1\ 083\ 883.$

[15] Following Day et al. (1997), we do not add programming effort to the overall tagging complexity. The sieving tool used in our experiment is now available to be reused for the annotation of other corpora, so in the long run, its implementation cost will be negligible anyway.

of the first letter being capitalized, plus a few rules to handle exceptions). The upper bound, in turn, is calculated by assuming that the sieving tool does not handle Proper Names.

Lower bound. The sieving tool was experimentally run over a corpus with 260 Ktokens ("exp-corpus", from this point, ftn 6). The following results were obtained: No tags: 64%; One tag: 16%; More than one: 20%.

From these values, it can be extrapolated that around 64% of our working corpus of 1 million tokens may be accurately tagged simply by running the sieving tool over it, i.e., without any increase in the complexity of the task of accurately annotating that corpus.

Given this tool is designed to tag a given token iff it assigns to it all its admissible tags, to the other portion of the corpus with tokens that received more than one tag, we refer as the directly detected ambiguity portion (20%). We refer to the portion of the corpus with tokens that received no tag as indirectly detected ambiguity portion (16%).

The tag to assign to each token in the directly detected ambiguity portion is selected from the tags already assigned by the sieving tool. To estimate the complexity of the subsequent disambiguation task, the different degrees of ambiguity involved should be considered. The distribution of tokens with several tags assigned to them is: 2 tags, 59.23%; 3, 29.14%; 4, 0.63%; 5, 0.09%; 6, 1.18%; MWU, 9.34%.

Most of the directly detected ambiguity involves 2 tags per token. As the number of tags per token increases, the frequency of such ambiguities decreases.[16] MWU is a special case of ambiguity where the elements in a sequence of tokens are individually or collectively tagged as a single MWU.

Getting back to the 1 million corpus, the above considerations imply that ca. 200 Ktokens (20%) can be annotated by picking the correct tag from the tags assigned by the sieving tool. For 29.14% of these 200 Ktokens, for instance, this requires picking one tag out of three, thus involving on average 2 sts. Repeating this calculation for each level of ambiguity, the complexity of accurately annotating the directly detected ambiguity portion in the working corpus is estimated as 346 360 sts.[17]

Turning to the indirectly detected ambiguity portion, our experiment shows that after running the sieving tool, ca. 160 Ktokens (16%) receive no tag. Given the tool exhausted the tags to be assigned to closed classes plus Proper Names,

[16] There is an odd increase in the frequency of ambiguities involving 6 tags due to two specific Portuguese forms, viz. como and nada. Both are very frequent and ambiguous: *como* receives the tags `INT`, `REL`, `CJ`, `PREP`, `ADV` and `V`, and *nada* receives `IN`, `DIAG`, `ADV`, `CN`, `ADJ` and `V`.

[17] $= 200 \times 10^3 \times (1.5 \times 0.5923 + 2 \times 0.2914 + 2.5 \times 0.0063 + 3 \times 0.0009 + 3.5 \times 0.0158 + 2 \times 0.0934) = 346\,360$. We assumed that on average two inspection steps are required to review potential multi-word lexical units.

every token in this portion is potentially ambiguous between three open classes: Adjective, Common Noun or Verb. The task of hand tagging is restricted now to picking one of these three tags.

With the tagged version of the exp-corpus used in the experiment, it is possible to determine that Common nouns are 49.8%, Verbs 36.7% and Adjectives 13.5% of the indirectly detected ambiguity portion. This allows to measure the complexity of the task of annotating the last 160 Ktokens of the working corpus, yet to be tagged. The three possible tags remaining are ranked according to the decreasing order of their frequencies in this portion: Nouns, Verbs, Adjectives. This implies that, for instance, tagging each Adjective requires 2 sts and annotating every Adjective in this indirectly detected ambiguity portion involves 43 200 sts.[18] Putting all the figures together, tagging the last 16% requires 261 920 sts.[19]

Collecting the values for both directly and indirectly detected ambiguity, the lower bound value for the complexity of tagging our working corpus with the sieving procedure is 608 280 sts.[20]

Upper bound. The upper bound is determined by assuming that the sieving tool is not prepared to handle Proper Names. As Proper Names are 7% of the exp-corpus, this implies that the indirectly detected ambiguity portion is enlarged now from 16% to 23% and the tags available to tag this portion are now four: Common Nouns, Adjectives, Verbs and Part of Name.[21] The exp-corpus allows also to know that Common nouns are 34.9%, Part of Names 29.9%, Verbs 25.7%, and Adjectives 9.5% of the indirectly detected ambiguity portion.

It is now possible to obtain the complexity for the indirectly detected ambiguity portion in the working corpus, with 230 Ktokens (23%). It amounts to 482 540 sts.[22]

Note that under this upper bound scenario, the value for the complexity of annotating the directly detected ambiguity portion is the same as the value obtained under in the lower bound scenario: the tokens tagged as Part of Names then are assigned only one tag thus being part of the 64% of the corpus correctly tagged with the sieving tool. Taking thus that value for the complexity of tagging the directly detected ambiguity portion together with the value just calculated above for the indirectly detected ambiguity portion, we obtain 828 900 sts[23] as the upper bound value. Taking the values for the upper and lower bounds of the sieve method, its complexity can be estimated as lying in the range 608–829 Ksts.

---

[18] $= 160 \times 10^3 \times 2 \times 0.135 = 43\,200$.

[19] $= 160 \times 10^3 \times (1 \times 0.498 + 2 \times 0.367 + 3 \times 0.135)$.

[20] $= 346\,360 + 261\,920 = 608\,280$.

[21] The tokens ambiguous between Proper Names and another tag have a residual value so it can be safely assumed that these 7% were displaced from the portion that received only one tag.

[22] $= 230 \times 10^3 \times (1 \times 0.349 + 2 \times 0.299 + 3 \times 0.257 + 4 \times 0.095)$.

[23] $= 346\,360 + 482\,540 = 828\,900$.

5 Discussion

The values of the upper and lower bounds of tagging complexity for the three procedures are compiled in the table below as well as the efficiency gains by each of them with respect to the others:

			scratch	t-t-r		sieve	
		Ksts	lb	ub	lb	up	lb
scratch	upper	20000	74.03	93.64	94.58	95.85	96.96
	lower	5194		75.51	79.13	84.04	88.29
t-t-r	upper	1272			14.78	34.83	52.20
	lower	1084				23.52	43.90
sieve	upper	829					26.66

When comparing even the best score of the from-scratch procedure with the worst results from the other procedures, the from-scratch method is confirmed as the least efficient one. Any of the other methods permits dramatic savings in terms of hand tagging effort: The train-tag-review allows to save at least 75.51% of the annotation effort, while the sieve method permits to save at least 84.04%.

The results are more instructive when it comes to compare the two more efficient procedures. In the worst case (the train-tag-review procedure's best score is compared with the sieve procedure's worst score), the latter permits to save almost one fourth (23.52%) of the annotation effort needed if the former method is adopted. If the procedures are compared on an equal footing (taking the best scores), the advantage of the sieve method over the train-tag-review one is larger: It permits to save well over one third (43.90%) of the annotation effort.

5.1 Invariance. While the metric for determining tagging complexity is independent of the corpora to be tagged and the human languages in which the data is encoded, the same may not be the case for the complexity scores of the different annotation procedures. Different corpora have different distributions for the frequencies of tags. When using one of the bootstrapping procedures to tag two corpora, it is likely that the two tasks show different complexity values. However, as for each tag, the fluctuation of its relative frequency with respect to different corpora tend to be limited[24] and it is much less than the difference between the complexity of the different methods, the conclusions drawn from the comparison exercise above concerning the ranking of tagging procedures are expected to remain basically valid for a wide range of different corpora.

The same invariance may not hold, however, when different languages are considered, especially if they belong to different language families. When considering generic, large-scale corpora from different languages that can be tagged with the same or approximate tag sets, the relative frequencies of POS tags may present considerable differences. If these differences are large enough to have an

[24] The sieving tool was used over a second corpus with 11.5 million tokens extracted from the corpus available at `http://cgi.portugues.mct.pt/cetempublico/`. The values obtained deviate at most 1 point from the values presented in Section 4.3.

impact on the ranking of the tagging methods according to their complexity now obtained, this is something that has to be empirically verified for each particular case.

5.2 Concluding remarks. While it is important to keep in mind that the ranking of fully accurate tagging procedures may not be independent of the particular language being considered, it is worth noting that the goal here is not to present a definitive ranking of such procedures valid for all languages (something conceivably not possible). Rather, the aim was to show that it is feasible to design an objective, standard metric to predict such ranking when different procedures, different corpora or different languages are taken into account; that the ranking produced is reliable for guiding one to opt for the most efficient procedure; and that the efficiency gains detected (and corresponding gains in terms of labor costs) appeared to be so dramatic that a decision on which procedure to opt for in each case should be considered with the help of such a metric.

Annex

Tag	Category	Freq. (%)	Cplx. (sts)	Tag	Category	Freq. (%)	Cplx. (sts)
CN	common noun	15.37	1	IN	indef. nominal	0.23	21
PNT	punctuation	14.45	2	DFR	fraction denom.	0.21	22
PREP	preposition	14.39	3	ORD	ordinal	0.16	23
V	verb	11.33	4	MTH	month	0.11	24
DA	def. article	11.27	5	WD	week day	0.07	25
PNM	part of name	6.87	6	STT	social title	0.06	26
ADV	adverb	5.29	7	ITJ	interjection	0.06	27
CJ	conjunction	4.77	8	INT	int. pronoun	0.06	28
ADJ	adjective	4.17	9	DIAG	dialogue	0.05	29
PTP	past participle	1.78	10	SYB	symbol	0.05	30
IA	indef. article	1.61	11	EADR	email address	0.03	31
REL	rel. pronoun	1.55	12	PADR	part of address	0.02	32
CL	clitic	1.50	13	MGT	magnitude	0.01	33
DGT	digit	0.86	14	PP	prep. phrase	0.01	34
DEM	demonstrative	0.84	15	DGTR	roman digit	0.00	35
PRS	pers. pronoun	0.71	16	LTR	letter	0.00	36
CARD	cardinal	0.61	17	NP	noun phrase	0.00	37
QD	quant. det.	0.61	18	EOE	end of enum.	0.00	38
POSS	possessive	0.59	19	UNIT	measure unit	0.00	39
GER	gerund	0.30	20				

REFERENCES

Banko, Michele & Eric Brill. 2001. "Scaling to Very Very Large Corpora for Natural Language Disambiguation". *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (*ACL'01*), 26-33. Toulouse, France.

Brants, Thorsten. 2000. "TnT — A Statistical Part-of-speech Tagger". *6th Applied Natural Language Processing Conference*, 224-231. Seattle, Washington.

Day, David, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson & M. Vilain. 1997. "Mixed-Initiative Development of Language Processing Systems". *Applied Natural Language Processing Conference* (*ANLP-97*), 348-355. Washington, D.C.

Voutilainen, A. 1999. "An Experiment on the Upper Bound of Interjudge Agreement". *9th Conf. of the European Chapter of the Association for Computational Linguistics* (*EACL'99*), 204-208. Bergen, Norway.

Index

A.
annotation
    annotation efficiency 183
    corpus annotation 175

P.
part-of-speech (POS)
    manual tagging 175
    tagging 175
    tagging efficiency 183