

Annotation Time Stamps — Temporal Metadata from the Linguistic Annotation Process

Katrin Tomanek, Udo Hahn

Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena
Fürstengraben 30, 07743 Jena, Germany
{katrin.tomanek|udo.hahn}@uni-jena.de

Abstract

We describe the re-annotation of selected types of named entities (persons, organizations, locations) from the MUC7 corpus. The focus of this annotation initiative is on recording the time needed for the linguistic process of named entity annotation. Annotation times are measured on two basic annotation units – sentences *vs.* complex noun phrases. We gathered evidence that decision times are non-uniformly distributed over the annotation units, while they do not substantially deviate among annotators. This data seems to support the hypothesis that annotation times very much depend on the inherent ‘hardness’ of each single annotation decision. We further show how such time-stamped information can be used for empirically grounded studies of selective sampling techniques, such as Active Learning. We directly compare Active Learning costs on the basis of token-based *vs.* time-based measurements. The data reveals that Active Learning keeps its competitive advantage over random sampling in both scenarios though the difference is less marked for the time metric than for the token metric.

1. Introduction

Cost awareness has not been a primary concern in most of the past linguistic annotation initiatives. Recently, strategies which strive for minimizing the annotation load have gained increasing attention. Selective sampling of the units to be annotated, such as Active Learning (Cohn et al., 1996), is certainly one of the most promising approaches. Still, when it comes to the empirical assessment of annotation costs even proponents of Active Learning make overly simplistic and empirically questionable assumptions, e.g., the uniformity of annotation costs over the number of linguistic units (typically tokens) to be annotated.

In this paper, we describe $MUC7_{\mathcal{T}}$, an extension of the MUC7 corpus (Linguistic Data Consortium, 2001), where we couple common named entity annotation metadata with a time stamp which indicates the time measured for the linguistic decision making process.¹ In $MUC7_{\mathcal{T}}$, annotation time meta data is available for sentences as well as for noun phrases as annotation units. The second part of the paper shows how this new resource can be applied in the context of effort leveraging annotation strategies.

The rest of the paper is structured as follows. Section 2. gives a detailed description of the annotation setting, describes how and on which unit level time measurements were taken, and evaluates the annotator’s performance in terms of inter-annotator agreement. Section 3. provides $MUC7_{\mathcal{T}}$ statistics, with the main finding that annotation time is subject to high variation. This supports our assumption of *non-uniform* time costs. We then apply the $MUC7_{\mathcal{T}}$ corpus for a cost-sensitive evaluation of a standard approach to Active Learning in Section 5. Finally, Section 6. concludes and points out additional application scenarios for the $MUC7_{\mathcal{T}}$ corpus.

¹These time stamps should not be confounded with the annotation of temporal expressions (TIMEX in MUC7).

2. Corpus Re-Annotation

2.1. Annotation Task Setup

Our re-annotation initiative targets the named entity annotations (ENAMEX) of the English part of the MUC7 corpus, *viz.* PERSONS, LOCATIONS, and ORGANIZATIONS. Temporal and number expressions (TIMEX and NUMEX) were deliberately ruled out. The annotation was done by two advanced students of linguistics with good English language skills. For consistency reasons, the original guidelines of the MUC7 named entity task were used.

MUC7 covers three distinct document sets for the named entity task. We used one of these sets to train the annotators and to develop the annotation design, and another one for the actual annotation experiment which consists of 100 articles reporting on airplane crashes. We split lengthy documents (27 out of 100) into halves so that they fitted in the screen of the annotation GUI without the need for scrolling. Still, we had to exclude two documents due to extreme over-length which would have required overly many splits. Our final corpus contains 3,113 sentences (76,900 tokens) (see Section 3. for more details).

Annotation time measurements were taken on two syntactically different *annotation units* of single documents: (a) complete sentences and (b) complex noun phrases. The annotation task was defined such as to assign an entity type label to each token of an annotation unit. The use of complex noun phrases (CNPs) as an alternative annotation unit is motivated by the fact that in MUC7 the syntactic encoding of named entity mentions basically occurs through nominal phrases. CNPs were derived from the sentences’ constituency structure using the OPENNLP parser (trained on PENNTREEBANK (Marcus et al., 1993) data) to determine top-level noun phrases.² To avoid overly long phrases, CNPs dominating special syntactic structures, such as coordinations, appositions, or relative clauses, were split up at

²<http://opennlp.sourceforge.net/>

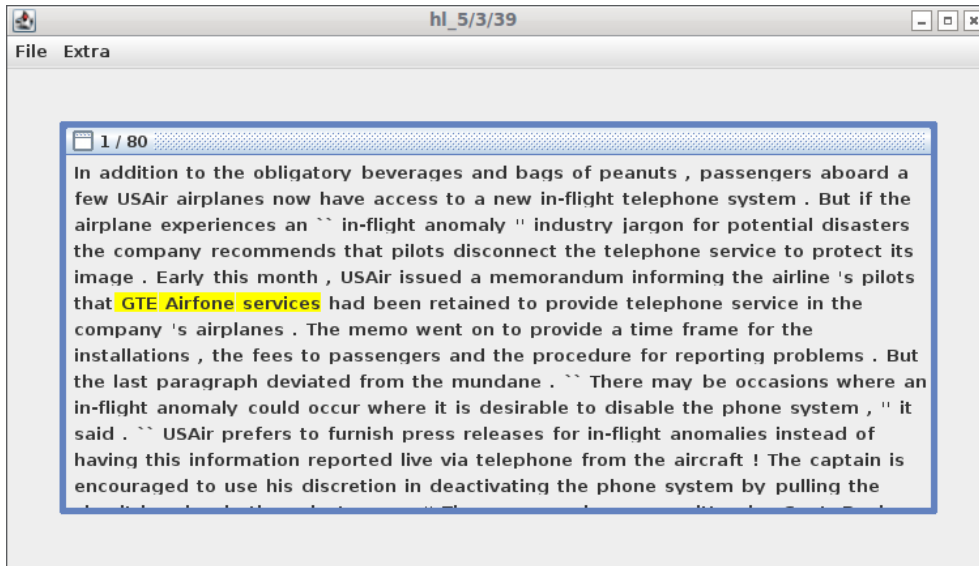


Figure 1: Screenshot of the annotation GUI showing an annotation example where the the complex noun phrase “GTE Airfone services” is highlighted for annotation.

discriminative functional elements (e.g., a relative pronoun) and these phrases were eliminated from further analysis.

An evaluation of our CNP extractor on ENAMEX annotations in MUC7 showed that 98.95% of all entities were completely covered by automatically identified CNPs. Incomplete coverage was mostly due to parsing errors.

While the annotation task itself was “officially” declared to yield annotations of named entity mentions within the different annotation units, we were nevertheless primarily interested in the time needed for making annotation decisions. For precise time measurements, single *annotation examples* were shown to the annotators, one at a time. An annotation example consists of the chosen MUC7 document with one annotation unit (sentence or CNP) selected and highlighted (yet, without annotation). Only the highlighted part of the document could be annotated and the annotators were asked to read only as much of the visible context surrounding the annotation unit as necessary to make a proper annotation decision. Figure 1 shows a screenshot of the annotation GUI.

To present the annotation examples to annotators and allow for annotation without extra time overhead for the “mechanical” assignment of entity types, our annotation GUI is controlled by keyboard shortcuts. This minimizes annotation time compared to mouse-controlled annotation such that the measured time reflects only the amount of time needed for taking an annotation decision.

In order to avoid learning effects for annotators on originally consecutive syntactic subunits, we randomly shuffled all annotation examples so that subsequent annotation examples were not drawn from the same document. Hence, annotation times were not biased by the order of appearance of the annotation examples.

Annotators were given blocks of either 500 CNP-level or 100 sentence-level annotation examples. They were asked to annotate each block in a single run under noise-free conditions, without breaks and disruptions. They were also in-

structed not to annotate for too long stretches of time to avoid tiring effects making time measurements unreliable. All documents were first annotated with respect to CNP-level examples within 2-3 weeks, with only very few hours per day of concrete annotation work. After completion of the CNP-level annotation, the same documents had to be annotated on the sentence level as well. Due to randomization and rare access to surrounding context during the CNP-level annotation, annotators credibly reported that they had indeed not remembered the sentences from the CNP-level round. Thus, the time measurements taken on the sentence level do not seem to exhibit any human memory bias.

Both annotators went through all annotation examples so that we have double annotations of the complete corpus.

2.2. Annotation Performance

To assess the quality of the performance of our annotators (henceforth called A and B), we compared their annotation results on 5 blocks of sentence-level annotation examples created in the training phase. Annotation performance was measured by Cohen’s κ coefficient on the token level and by determining the entity-segment *F*-score against MUC7 annotations. The annotators A and B achieved $\kappa_A = 0.95$ and $\kappa_B = 0.96$, and $F_A = 0.92$ and $F_B = 0.94$, respectively.³ Moreover, they exhibit an inter-annotator agreement of $\kappa_{A,B} = 0.94$ and an averaged mutual F-score of $F_{A,B} = 0.90$.

These numbers reveal that the task was well-defined and the annotators had sufficiently internalized the annotation guidelines. Although we were not specifically interested in the annotations itself, high annotation performance is required for valid time measurements. Figure 2 shows the annotators’ scores against the original MUC7 annotations for the 31 blocks of sentence-level annotations. Kappa scores

³Entity-specific F-scores against MUC7 annotations for A and B are 0.90 and 0.92 for LOCATION, 0.92 and 0.93 for ORGANIZATION, and 0.96 and 0.98 for PERSON, respectively.

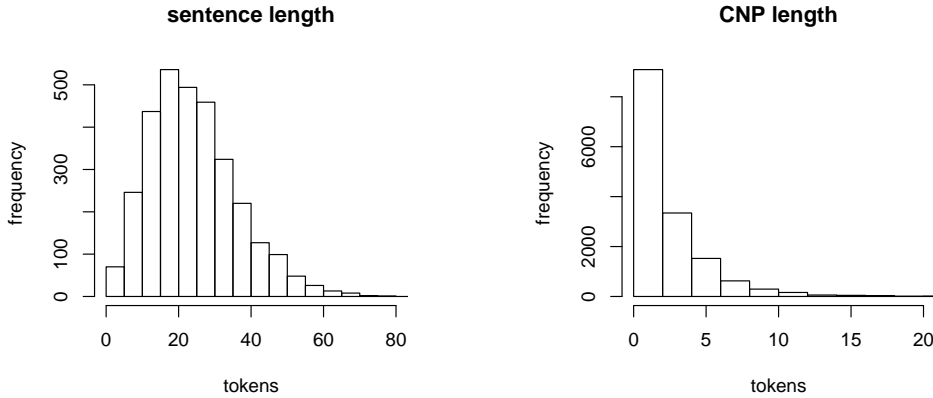


Figure 3: Length distribution of sentences and CNPs.

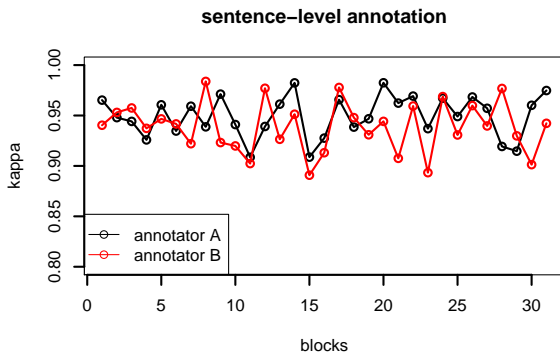


Figure 2: Average kappa coefficient per block.

range from $\kappa = 0.89$ to $\kappa = 0.98$ and annotation performance is similar for both annotators. Annotation performance shows that they consistently found a block either rather hard or easy to annotate. Moreover, annotation performance appears stationary – no general trend in annotation performance over time can be observed.

3. MUC_{7 \mathcal{T}} Corpus Statistics

MUC_{7 \mathcal{T}} comprises 3,113 sentences which amount to 76,900 tokens. About 60% of all tokens are covered by CNPs showing that sentences are made up from CNPs to a large extent. Still, removing the non-CNP tokens markedly reduces the amount of tokens to be considered for entity annotation. CNPs cover slightly less entities (3,937) than complete sentences (3,971). This marginal loss is due to the incomplete coverage of the CNP extractor. Table 1 summarizes statistics on the time-stamped MUC7 corpus.

On the average, sentences have a length of 24.7 tokens, while CNPs are rather short with 3.0 tokens, on the average. However, CNPs vary tremendously in their length, with the shortest ones having only one token and the longest ones (mostly due to parsing errors) spanning over 30 (and more) tokens. Figure 3 depicts the length distribution of sentences and CNPs showing that a fair portion of CNPs have less than five tokens, while the distribution of sentence lengths almost follows a normal distribution in the interval $[0, 50]$.

sentences	3,113
sentence tokens	76,900
chunks	15,203
chunk tokens	45,097
entity mentions in sentences	3,971
entity mentions in CNPs	3,937
sentences with entity mentions	63%
CNPs with entity mentions	23%

Table 1: Descriptive statistics of MUC_{7 \mathcal{T}} .

While 63% of all sentences contain at least one entity mention, only 23% of the CNPs contain entity mentions. These statistics show that CNPs are generally rather short and a large fraction of CNPs do not contain entity mentions at all. We may hypothesize that this observation will be reflected by annotation times.

4. Time Measurements

The annotation process should not be subject to learning effects but be stationary instead. This requirement holds especially for our time measurements, otherwise the use of the time data for learning cost models or evaluating sampling strategies might lead to questionable results.

Figure 4 shows the average annotation time per block (CNPs and sentences). Considering the CNP-level annotations, we found a slight learning effect for annotator B during the first 9 blocks and no learning effect at all for annotator A. After this ‘calibration’ phase for annotator B, both annotators are approximately on a par regarding the annotation time. For the sentence-level annotations, no learning effect at all could be identified because both annotators yield similar annotation times per block. So, with the exception of block one to nine for annotator B, time measurements are also stationary.

Figure 4 also shows that there is quite some variation in the average annotation time per blocks, especially in the sentence-level annotation setting. While the annotation of the sentences in blocks 25 and 26 require about 6 seconds on average, only about 4.5 seconds are required in blocks 28 and 29. As for CNP-level annotations, this variation is

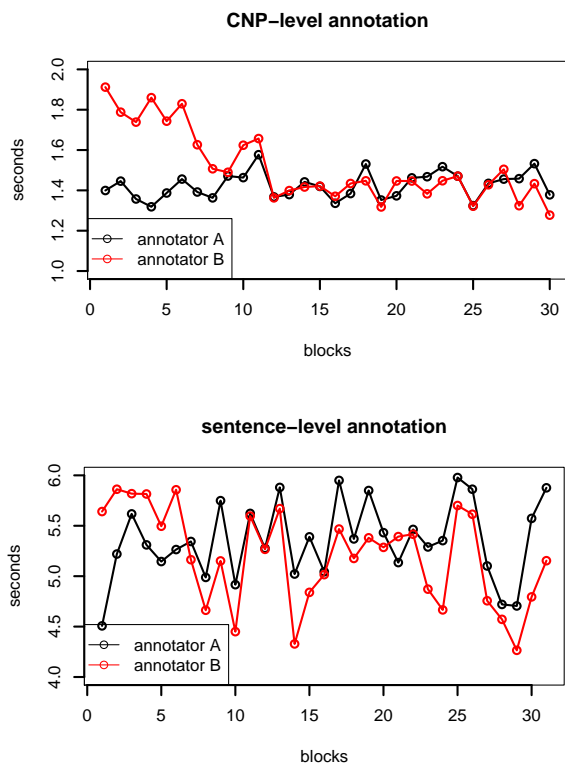


Figure 4: Average annotation times per block. Annotator B exhibits a learning effect within the first 9 blocks of CNP-level annotation.

less pronounced when considering complete blocks.

For further investigation into the variation of annotation times, Figure 5 shows the distribution of annotator A’s CNP-level annotation times for block 20. A’s average annotation time on this block amounts to 1.37 seconds per CNP, the shortest duration being 0.54, the longest one running up to 10.2 seconds. The figure provides ample evidence for an extremely skewed time investment for coding CNPs.

In summary, both annotation performance and annotation time are basically stationary which allows an independent interpretation of single time measurements. However, time and performance plots also clearly reveal that some blocks were generally harder or easier than others because both

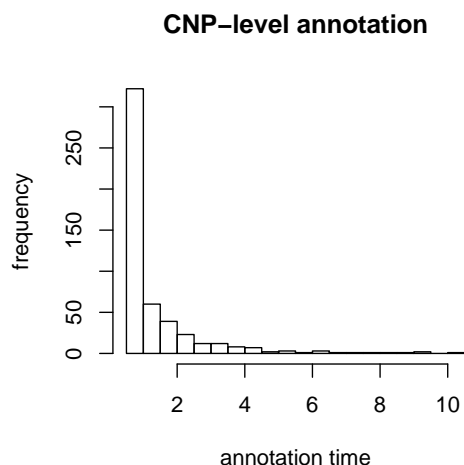


Figure 5: Distribution of annotation times.

annotators operated on both metrics in tandem – they both spent more time and reached lower Kappas on harder ones than on easier ones.

5. Application of $MUC7_T$ to Evaluate Selective Sampling Strategies

Section 4. gave strong evidence that the assumption of uniform annotation costs per annotated unit is untenable. Accordingly, methods which aim at making the inherently costly process of language resource creation more economical through intelligent selection of the items to be manually annotated should be based on valid cost criteria. As things stand, we have to give up over-simplifying cost models which rely on counting the number of tokens in favor of empirically more adequate annotation time metrics.

As an example, we now will discuss Active Learning (AL), an approach to reduce annotation efforts, from the perspective of alternative cost models. In the AL scenario, the learner is in control of the data to be chosen for training. Labels are only requested from a human annotator for such examples which are (estimated) to have a high utility for the training process. For a wide range of NLP problems to which supervised machine learning methods were applied, it has already been shown that AL can indeed dramatically decrease the number of training examples needed to yield a certain target performance (Engelson and Dagan, 1996; Ngai and Yarowsky, 2000; Ringger et al., 2007; Tomanek et al., 2007). However, all arguments concerning cost efficiency were based on token counts in these studies.

In the following, we compare the standard token-based measurement of efficiency against a metric based on annotation time costs by using the $MUC7_T$ corpus. We apply a standard AL approach known as Uncertainty Sampling (Cohn et al., 1996) where the utility of an example is based on the model’s uncertainty in its prediction. Algorithm 1 formally describes the AL procedure. In each AL iteration, b examples are selected, handed to the annotator for labeling, and then added to the set of labeled training data \mathcal{L} which feeds the classifier for the next training round.

Algorithm 1 Uncertainty Sampling AL

Given:

\mathcal{L} : set of labeled examples, \mathcal{P} : set of unlabeled examples
 b : number of examples to be selected in each iteration
 $u(x)$: utility function

Algorithm:

loop until stopping criterion is met

1. learn model θ from \mathcal{L}
2. $b' = 0$; while $b' < b$
 - select example: $x^* = \operatorname{argmax}_{x \in \mathcal{P}} u(x)$
 - query annotator for label y^* for x^*
 - move example: $\mathcal{P} = \mathcal{P} \setminus x^*$, $\mathcal{L} = \mathcal{L} \cup (x^*, y^*)$
 - $b' = b' + 1$

return \mathcal{L}

The utility of an unlabeled example $x \in \mathcal{L}$ is calculated as

$$u(x) = 1 - \max_{y' \in \mathcal{Y}} P_{\theta}(y'|x)$$

where $P_{\theta}(y'|x)$ is the confidence of model θ that y' is the correct label. While many different utility functions for AL

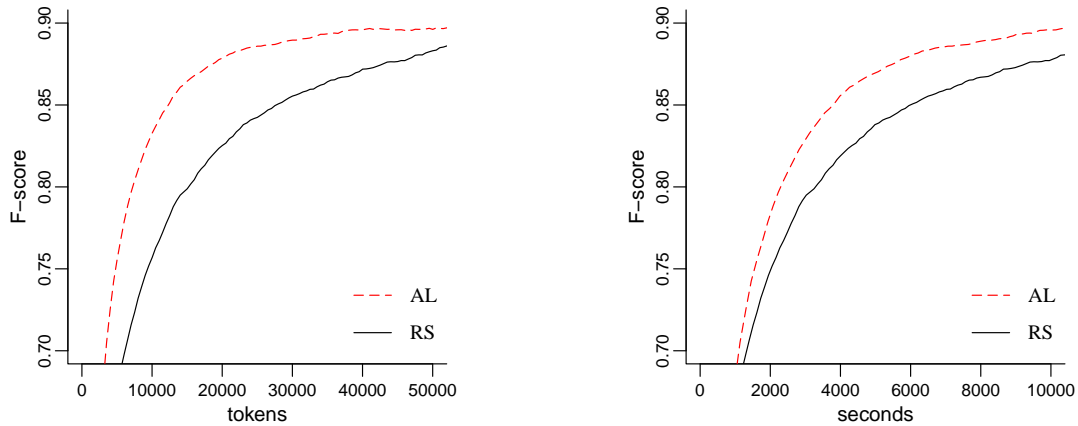


Figure 6: Learning curves for AL and RS evaluated against number of tokens and true annotation time as available through the $MUC7_{\mathcal{T}}$ corpus.

have been proposed (Settles and Craven, 2008), we deliberately chose this straight-forward utility function for reasons of simplicity.

We apply Conditional Random Fields (CRFs) (Lafferty et al., 2001) as our learning algorithm for the NER task and make use of a rich set of standard features for statistical NER. In this experiment, we consider complete sentences as an appropriate example grain size so that in each AL iteration b sentences are selected for manual annotation.

Results reported are averages over 20 independent runs. For each run, the $MUC7_{\mathcal{T}}$ corpus was randomly split into a pool from which AL selects (90% of the sentences) and an evaluation set used to plot the learning curve (remaining sentences). AL was started from 20 randomly selected sentences; 20 sentences were selected in each AL iteration. The annotation time assessment is taken from the sentence-level time stamp metadata of the $MUC7_{\mathcal{T}}$ corpus (we chose annotator A’s time stamps here).

Figure 6 shows the performance of this AL approach by learning curves which describe model performance as *a*) a function of corpus size and *b*) a function of the annotation time. Comparing AL with the random selection (RS) of examples, which is still the standard procedure in most annotation campaigns, it is evident that AL is much more efficient than RS both in terms of corpus size as well as annotation time. In terms of tokens, an F-score of 89% is reached by RS after manual annotation of 60,015 tokens, AL requires only 32,513 tokens which is a decrease of annotation efforts of about 46%. When looking at the annotation time needed to achieve the same F-score level, AL is still much more efficient than RS, consuming only 8,728 seconds instead of about 13,000 seconds — a real saving of annotation time of about 33%. Obviously, AL does not only decrease the required corpus size but can indeed reduce the necessary annotation time considerably.

However, comparing both cost metrics, AL still does much better when merely tokens are counted and performs worse when annotation time is taken into consideration. We claim, however, that the time annotators spend doing their job is a more realistic metrical unit than the number of tokens they deal with. Fortunately, even under this more

realistic ‘budgetary’ perspective the advantages of Active Learning are preserved and an efficient alternative to random sampling exists.

Claire et al. (2005) and Hachey et al. (2005) also found that the actual sampling efficiency of an AL approach depends on the cost metric being applied. They studied how sentences selected by AL affected the annotators’ performance both in terms of the time needed and the annotation accuracy achieved. They found that selectively sampled examples are, on the average, more difficult to annotate than randomly sampled ones. This observation, for the first time, questioned the widespread practice that all annotation examples can be assigned a uniform cost factor. It also raises another interesting and open issue, *viz.* whether examples with a high utility for classifier training are, at the same time, also cognitively more demanding, e.g., due to their intrinsic linguistic or conceptual complexity.

6. Conclusions

This paper proposed a new breed of metadata for a linguistic corpus, *viz.* information on the time it takes to add certain linguistic annotations, such as NER in our case. For this purpose, we have created a time-stamped version of $MUC7$ entity annotations, $MUC7_{\mathcal{T}}$.

An analysis of the time stamps recorded in $MUC7_{\mathcal{T}}$ provides ample evidence for the intuitive assumption that the time needed to annotate a particular unit varies considerably (independent from single annotators). Moreover, we showed how such a corpus can be used to assess, in a realistic scenario, the sampling efficiency of AL strategies where the goal should be not only to decrease the corpus size but even more so to decrease the annotation effort in terms of actual time needed to perform the annotation task.

To make AL more cost-conscious, estimated annotation time may be incorporated into the selection process, so that examples which are highly informative, but come with extremely high costs in annotation time, are ignored. In this context another issue arises, *viz.* the prediction of annotation time in real applications where time stamps are naturally not available. The availability of annotation time information on linguistically well-motivated and fine-grained

