Asif Ekbal, Francesca Bonin, Sriparna Saha, Egon Stemle, Eduard Barbu,Fabio Cavulli, Christian Girardi, Massimo Poesio

Rapid Adaptation of NE Resolvers for Humanities Domains using Active Annotation

The entities mentioned in collections of scholarly articles in the Humanities (and in other scholarly domains) belong to different types from those familiar from news corpora, hence new resources need to be annotated to create supervised taggers for tasks such as NE extraction. However, in such domains there is a great need for making the best use possible of the annotators. One technique designed for this purpose is **active annotation**. We discuss our use of active annotation for annotating corpora of articles about Archaeology in the Portale della Ricerca Umanistica Trentina.

1 Introduction

Many of the entities mentioned in collections of scholarly articles in subjects such as Archaeology, History, or History of Art do not belong to the types found in the news corpora on which Computational Linguistics work has focused, such as the MUC and ACE corpora. For instance, the most important entity types found in archaeological texts are `Culture`, `Site`, and `Artefact`. In some such domains, even if more familiar types such as `Person` play an important role, it is essential to distinguish between their subtypes. E.g., in History of Art articles, it is not enough to classify an entity as a `Person`; it is also crucial to recognize if a particular individual was a `Painter`, a `Sculptor`, an `Architect`, etc. Hence, dedicated resources need to be created to train Named Entity (NE) recognizers for these domains; training on news corpora is of limited use to extract semantic content from such articles.

However, creating resources is always expensive, and Humanities projects tend not to have lots of funding for these purposes. In addition, collections of articles in the Humanities tend to be fairly small.It is therefore essential to use the limited funding available wisely, and to maximise the benefit to be obtained from the data. In other words, this is a domain for which **active learning** techniques (Settles, 2009), already used for NE tagging by, e.g., Vlachos (2006), seem ideally suited.

In this paper we discuss our work on using active learning for NE annotation of a corpus of scholarly articles in the Humanities being created in support of the creation of the Portale della Ricerca Umanistica Trentina, whose aim is to give scholars and the general public entity-, spatial-, and temporal-indexing based methods to access the many different collections of scholarly articles in the Humanities held by private and public collections in Trentino. After a brief introduction to the Portale della Ricerca Umanistica Trentina and the corpus under creation in Section 2, we introduce our approach to combining active learning with crf-based ne tagger in Section 3, and the results obtained in Section 4.

2 The Portale della Ricerca Umanistica Trentina

2.1 Aims

The Portale della Ricerca Umanistica Trentina (Humanities Research Portal, pru) (Poesio et al., 2011a) is a pilot project to set up a one-stop search facility for repositories of scholarly articles and other types of publications in the Humanities held by digital libraries, museums and archives in Trentino. The portal will use content extraction techniques to automatically extract citations and semantic metadata including temporal, spatial, and entity references from the publications in those repositories. This information will then be used to offer visitors to the portal two main functionalities: **content-based search and browsing** and **semantic uploading**.

Besides standard keyword-based search, the pru will also offer **entity-based search**. Two types of browsing will be possible: **spatial** and **temporal** browsing. Entity search allows users to retrieve all documents that discuss a particular entity irrespective of the way it's called–e.g., all Archaeological documents that discuss sites in which a particular shellfish was found irrespective of whether it's called in the document *Spondylus sp.* or *Spondilo*. Spatial browsing allows users to retrieve the publications that mention a particular locality in Trentino by visualizing a map of Trentino and clicking on the appropriate location. Temporal browsing (currently under development) will allow users to retrieve all historical articles discussing a particular period.

These novel types of searching and browsing will be supported by a **semantic upload function**: registered scholars and / or curators of the collections will be able to upload publications that will then be processed by the pru pipeline discussed below to automatically extract both metadata and information about the publication to be inserted in the catalogue of the repository after being checked by the curator.

The first repository whose documents have been made accessible through the pru is the collection of articles in the Archaeological domain in the apsat / alpinet digital library. We are currently working on indexing other repositories as well.

2.2 The apsat / alpinet Portal and Collection

The apsat / alpinet portal is a pilot Spatial Humanities project developed by the University of Trento's "B. Bagolini" Lab and allowing scholars to visualize Archaeological sites in the Alps through a Web GIS interface, through which Scholars can examine an area in general to find which sites are present, or look in detail at the features of a particular site. Through the portal, scholars can also access Archaeological articles about these sites, either through keywords or through the Web GIS interface.

Among the holdings of the portal is the complete collection of the journal *Preistoria Alpina* published by the Museo Tridentino di Scienze Naturali. We will focus on this collection in the present work. The collection is multilingual, containing articles written in English, French, German and Italian; in fact, as typical of the Humanities, many

NE type	Details
Culture	Artefact assemblage characterizing a group of people in a specific time and place
Site	Place where the remains of human activity are found (settlements, infrastructures, cimiteries, production site, ...)
Artefact	Objects created or modified by men (tools, vessels, ornaments, ...)
Ecofact	Biological and environmental remains different from artefacts but culturally relevant (e.g., *Spondylus*)
Feature	Remains of construction or maintenance of an area related with dwelling activities (fire places, post-holes, pits, channels, walls, ...)
Location	geographical reference
Time	historical periods
Organization	association (no publications)
Person	human being discussed in the text (e.g., Ötzi the Iceman, Pliny the Elder, Caesar)
Pubauthor	author in bibliographic references
Publoc	publication location
Puborg	publisher
Pubyear	publication year

Tabelle 1: Annotation scheme for Named Entities in the Archaeology Domain

articles are themselves multilingual, in that they contain, in addition to text in the main language, an abstract, keywords, and occasionally captions in a second language, often but not always English.

2.3 A Structure-Sensitive, Multilingual Pipeline

The articles to be made accessible through the PRU are processed by a pipeline that tokenizes, POS-tags, and NE tags the text in order to extract semantic indices (Poesio et al., 2011b). The pipeline, accessible as a Web service, is based on the TEXTPRO pipeline[1] (Pianta et al., 2008), and has two distinguishing features.

First, it is **structure sensitive**, in the sense that it includes a module that identifies the structure of a document to find citations and the like, in the manner of the FlyBase pipeline (Briscoe, 2011). Second, it is **constituent-level multilingual**, in that each constituent of the document structure is first run through a language identifier in order to find which version of the TEXTPRO system should be run on that constituent. (English and Italian are supported at the moment.) The first version of the pipeline included the default TEXTPRO NE tagger, ENTITYPRO, trained to recognize the standard ACE entity types. The objective of this work was to create a corpus that could be used to train a new NE tagger able to recognize the relevant entities in the APSAT / ALPINET collection.

2.4 Annotation Scheme for the apsat / alpinet collection

The most important NE types for the domain, identified in collaboration with the domain experts from the Bagolini Lab, are shown in Table 1.

Two broad classes of entities were identified on the basis of the types of queries that may be performed: entities that are part of what may be considered the content matter of the article (sites, cultures, individuals, names of ecofacts found in sites such as *Spondylus*), and entities that are part of the bibliographical references (e.g., authors of papers cited, year of publication, etc.). One of the most interesting aspects of these

[1] http://textpro.fbk.eu/

data is the prevalence of underspecified references. For instance, the term *Fiorano* refers to a culture from the Ancient Neolithic, that takes its name from the site *Fiorano*, which in turn is named from *Fiorano* Modenese in Emilia; for many uses of this term, it is impossible to tell which sense is intended. Possible solutions to this problem are to develop a system for underspecified typing like the `GPE` type in the ACE annotations[2] or guidelines forcing one interpretation. For the moment, coders have been asked to tag such cases as `underspecified`; we intend to return to the issue discussing options with the Archaeology experts, and develop a scheme / carry out agreement studies then.

3 Active Annotation and Conditional Random Fields

In this Section we first briefly review the notion of active annotation and the Conditional Random Fields approach to supervised learning we used to train our NER system, before introducing the approach to selecting the most informative samples we adopted.

3.1 Active Annotation

Active annotation–the term introduced by Vlachos (2006) to refer to the application of active learning (Settles, 2009) to corpus creation–is becoming a popular annotation technique because it can lead to drastic reductions in the amount of annotation that is necessary for training a highly accurate statistical classifier. In the traditional, random sampling approach, unlabeled data is selected for annotation at random. In contrast, in active learning, the most useful data for the classifier are carefully selected. In a typical active learning setup, a classifier is trained on a small sample of the data (usually selected randomly), known as the **seed** examples. The classifier is subsequently applied to a pool of unlabeled data with the purpose of selecting additional examples that the classifier views as informative. The selected data is annotated and the cycle is repeated, allowing the learner to quickly refine the decision boundary between the classes.

The key question in this approach is how to determine the samples that will be most useful to the classifier. A number of techniques have been proposed, ranging from choosing the sample on which the classifier trained on the seeds is less certain, to a variety of entropy-based approaches (Vlachos, 2006; Settles, 2009). We discuss our approach after first introducing the supervised training method we chose.

3.2 Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are undirected graphical models, a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, these CRFs can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training, and are fast becoming the preferred method for NE tagging.

[2] (Buitelaar, 1998) is the earliest and possibly one of the most developed versions of this approach.

CRFs are used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $s =< s_1, s_2, \ldots, s_T >$ given an observation sequence $o =< o_1, o_2, \ldots, o_T >$ is:

$$P_\wedge(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

where $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k, is to be learned via training. The values of the feature functions may range between $-\infty, \ldots + \infty$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_o = \sum_s \exp(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

which as in HMMs, can be obtained efficiently by dynamic programming.

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L_\wedge = \sum_{i=1}^{N} \log(P_\wedge(s^{(i)}|o^{(i)})) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2},$$

where $\{< o^{(i)}, s^{(i)} >\}$ is the labeled training data. The second sum corresponds to a zero-mean, σ^2 -variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set parameters λ to maximize the penalized log-likelihood using Limited-memory BFGs (Sha and Pereira, 2003), a quasi-Newton method that is significantly more efficient, and which results in only minor changes in accuracy due to changes in λ.

When applying CRFs to the NER problem, an observation sequence is a token of a sentence or document of text and the state sequence is its corresponding label sequence. A feature function $f_k(s_{t-1}, s_t, o, t)$ has a value of 0 for most cases and is only set to be 1, when s_{t-1}, s_t are certain states and the observation has certain properties. We have used the C^{++} based CRF^{++} package, version 0.54[3], a simple, customizable, and open source implementation of CRF for segmenting or labeling sequential data.

3.3 Active Annotation with CRF

The main steps of the active annotation approach we followed are shown in Figure 1.

A feature vector consisting of the features described in the following Section is extracted for each word in the NE tagged corpus. Now, we have a training data in the form (W_i, T_i), where, W_i is the i^{th} word and its feature vector and T_i is its out-

[3] http://crfpp.sourceforge.net

```
Step 1:  Evaluate the system on the gold standard test data.
Step 2:  Test on the development data and calculate the conditional probabilities of all
         the output classes.
Step 3:  Compute the confidence interval (CI) between the two most probable classes
         for each token.
Step 4:  If CI is below the threshold value (set to 0.1 and 0.2) then
         Step 4.1: Add the NE token along with its sentence identifier and CI in a list of
            effective sentences, selected for active annotation (named as EA).
Step 5:  Sort EA in ascending order of CI.
Step 6:  Select the top most 10 sentences.
Step 7:  Remove the 10 sentences along with the preceding one and following one sentences
         from the development set.
Step 8:  Add the sentences to the training set.
Step 9:  Retrain the CRF classifier and evaluate with the test set.
Step 10: Repeat steps 2-9 until the performance in two consecutive iterations be same.
```

Abbildung 1: Main steps of the proposed active learning technique

put tag. We consider various combinations from the set of feature templates specified by:

F_1 ={$w_{i-m}, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_{i+n}$; Combination of w_{i-1} and w_i; Combination of w_i and w_{i+1}; Feature vector consisting of root word, prefix and suffix, PoS, first word, infrequent word, digit, content words, and capitalization of w_i; B}

where B denotes the bi-gram template that calculates all the feature combinations of the current and previous tokens. The CRF is trained with the above-mentioned feature set and evaluated on the gold standard test set. For CRF training, we set the following parameter values: regularization parameter (a): default setting, i.e. L2; soft-margin parameter (c): trades the balance between overfitting and underfitting (default value); and cut-off threshold for the features (f): uses the features that occurs no less than its value in the given training data (set to 1, i.e. all the features that appear at least once in the training dataset is considered). We varied the context within the previous two and next two words. New sentences are chosen from the development set and added to the initial training set using the following selection method.

For each token of the dataset containing additional data to annotate, our CRF classifier outputs the confidence values (conditional probabilities) of each class. Our proposed selection criterion is to choose the token for which the differences between the confidence values of the most probable two classes is smaller– the hypothesis being that items for which this difference is smaller are those of which the classifier is less certain. A threshold on the confidence interval is defined, and at each iteration we select for further annotation the sentences in the 'extension' dataset containing such items, have the annotators label them, and add them to training.

We tested two ways of adding to the training set: either (i) add only the current sentence that contains the most informative example, or (ii) add the current sentence

Set	# token	# NEs
Training	20,739	2,611
Development	5,292	622
Test	11,534	1,582

Tabelle 2: Statistics about the training, development and test sets

along with the previous one and next one sentences. Thus, in each iteration, we add either 10 or 30 sentences to the training set. The iteration stops when the performance in two consecutive iterations doesn't change.

4 Annotation Experiments

4.1 Datasets

In order to train and evaluate NE taggers for the domain, a small collection of papers from the journal *Preistoria Alpina* was annotated. 11 articles from the journal, for a total of around 50,000 tokens, were annotated according to the scheme in Section 2.4. Of these, five articles were randomly chosen as training set, three as test set, and three articles for active annotation and development. Some statistics about the training, development and test tests are shown in Table 2.

Basic NE tags were converted into the BIO format, where B–, I– and O– denote the beginning, inside and outside tokens of NEs. For example, the name *le conchiglie* gets tagged as *le*/B-Ecofact *conchiglie*/I-Ecofact.

4.2 Named Entity Features

We use the following main set of features, which are domain as well language independent in nature, and automatically extracted without the help of any domain dependent resources and/or language specific rules. We also compared these results with the results obtained by adding information extracted from a gazetteer.

1. Context words: These are the preceding and succeeding words of the current word. This is based on the observation that surrounding words carry effective information for the identification of NEs.

2. Word suffix and prefix: Fixed length (say, n) word suffixes and prefixes are very effective to identify NEs and work well for the highly inflective Indian languages. Actually, these are the fixed length character strings stripped either from the rightmost or from the leftmost positions of the words. If the length of the corresponding word is less than or equal to $n-1$ then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. This feature is included with the observation that NEs share some common suffixes and/or prefixes. Here, we consider prefixes and suffixes of length upto 3 characters.

3. First word: This is a binary valued feature that checks whether the current token is the first word of the sentence or not. We consider this feature with the observation that the first word of the sentence is most likely a NE.

4. Word length: We define a binary valued feature that fires if the length of w_i is greater than a pre-defined threshold. Here, the threshold value is set to 5. This feature captures the fact that short words are likely not to be NEs.
5. Infrequent word. A list is compiled from the training data by considering the words that appear less frequently than a predetermined threshold. The threshold value depends on the size of the dataset. Here, we consider the words having less than 10 occurrences in the training data. Now, a feature is defined that fires if w_i occurs in the compiled list. This is based on the observation that more frequently occurring words are rarely the NEs.
6. Capitalization: This is a binary valued feature that determines whether the word starts with a capital letter or not. This feature captures the fact that capitalized words are most likely NEs.
7. Part-of-Speech (PoS) information: PoS information of the current and/or the surrounding tokens(s) extracted using TextPro were used for NE identification.
8. Word normalization: We use a normalization feature clustering the words that have similar structures. This feature indicates how a target word is orthographically constructed. Word shapes refer to the mapping of each word to their equivalence classes. Here each capitalized character of the word is replaced by 'A', small characters are replaced by 'a' and all consecutive digits are replaced by '0'. For example, *Dalla* is normalized to *Aaaaa*, *123* is normalized to *0* and *1993* is also normalized to *0*.
9. Root word: The stems of the wordforms, extracted using TextPro.
10. Digit features: Several digit features are defined depending upon the presence and/or the number of digits and/or symbols in a token. These features are digitComma (token contains digit and comma), digitPercentage (token contains digit and percentage), digitPeriod (token contains digit and period), digitSlash (token contains digit and slash), digitHyphen (token contains digit and hyphen) and digitFour (token consists of four digits only).
11. Content words in global context: This feature is based on global contextual information. We consider all unigrams in contexts $w_{i-3}^{i+3} = w_{i-3} \dots w_{i+3}$ of w_i (crossing sentence boundaries) for the entire training data. We convert tokens to lower case, remove stopwords, numbers and punctuation symbols. We define a feature vector of length 10 using the 10 most frequent content words. Given a classification instance, the feature corresponding to token t is set to 1 iff the context w_{i-3}^{i+3} of w_i contains t.

4.3 Results

We trained a CRF model with the feature set mentioned in Section 4.2. We conducted a number of experiments with the various context sizes within the context window of $w_{i-2}, \dots, w_{i+2}$, and the feature template as mentioned in Section 3.3. We observed the best performance with the context of w_{i-1}, w_i, w_{i+1}, and thus only report its results. The best configuration is obtained by tuning the system on the development data. The system is evaluated using the evaluation metrics of standard recall, precision and F-measure. We used strict matching criteria, i.e. the system is given full credit only if the predicted labels of all the tokens of a NE is same as that of the gold labels.

Iteration	Threshold=0.1			Threshold=0.2			Baseline (random)		
number	r	p	F	r	p	F	r	p	F
1	63.02	65.48	64.23	64.32	67.83	66.03	64.64	66.35	65.47
2	64.73	67.11	65.90	65.84	68.81	67.29	64.21	65.99	65.09
3	65.08	67.92	66.47	66.10	69.6	67.81	65.40	66.90	66.14
4	65.66	68.41	67.01	66.80	70.09	68.41	65.86	67.73	66.78
5	66.82	69.62	68.19	67.68	70.92	69.27	65.54	67.25	66.39
6	67.31	70.06	68.66	68.26	70.26	69.24	65.66	67.25	66.44
7	67.63	70.31	68.94	68.26	70.54	69.38	65.77	67.41	66.58
8	67.63	70.31	68.94	68.26	70.54	69.38	66.90	68.56	67.72
9	67.86	70.57	69.19	68.83	70.99	69.89	67.19	68.90	68.04
10	67.86	70.57	69.19	68.83	70.99	69.89	67.19	67.90	68.04

Tabelle 3: Evaluation results of active learning with (a) threshold=0.1 (b) threshold=0.2 (c) random selection. Here, 'r': recall, 'p': precision, 'F': F-measure (we report percentages)

Iteration	Threshold=0.1					Threshold=0.2				
number	r	p	F	#sentence added	#NE added	r	p	F	#sentence added	#NE added
1	67.51	66.93	67.18	27	113	65.52	68.93	67.18	27	113
2	66.08	67.29	65.65	23	115	66.08	69.29	67.65	23	115
3	66.46	69.36	67.88	24	118	66.46	69.36	67.88	24	118
4	67.29	70.08	68.66	25	123	67.29	70.08	68.66	25	123
5	68.87	71.24	70.04	19	68	68.87	71.24	70.04	19	68
6	69.19	71.19	70.18	8	16	68.86	71.57	70.19	17	35
7	69.19	71.19	70.18	1	3	69.51	71.47	70.48	3	5
8	69.19	71.19	70.18	0	0	69.51	71.47	70.48	0	0
9	69.19	71.19	70.18	0	0	69.51	71.47	70.48	0	0
10	69.19	71.19	70.18	0	0	69.51	71.47	70.48	0	0

Tabelle 4: Evaluation results of active learning with (a) threshold=0.1 (b) threshold=0.2 by including gazetteer based features (we report percentages)

We experimented with the selection criteria that not only adds the current sentence but also adds the surrounding sentences (the preceding and the following sentences). We experiment with this selection with the intuition that wider context could give more useful information to the statistical classifier. For selecting the candidates of annotation, we determine the appropriate confidence thresholds from the development set.

The results of the proposed active learning technique with the confidence threshold of 0.1 are presented in Table 3. Here, the 10 most effective sentences and their preceding one and following one sentences are removed from the development set and added to the training set. The highest performance obtained with this method are recall, precision and F-measure values of 67.86%, 70.57% and 69.19%, respectively. This result is obtained at the ninth iteration and does not improve in the next iteration.

The results with a threshold of 0.2 are also shown in Table 3. The table shows that this threshold results in a better performance than with a threshold of 0.1: we obtained recall, precision and F-measure values of 68.83%, 70.99% and 69.89%, respectively.

The results of the baseline model, where in each iteration 10 sentences with their preceding and following ones are randomly chosen from the development set and added to training set, are shown in Table 3. Recall, precision and F-measure values of 67.19%, 67.90%, and 68.04%, respectively. This is lower in comparison to our proposed approach by 1.64, 2.09 and 1.85 percentage of recall, precision and F-measure values, respectively.

In our next experiment we used two gazetteers for the types `SITE` and `CULTURE` extracted from the ALPINET / APSAT database and containing 2,078 and 98 wordforms,

Class	Bound. Id. Error %
Artefact	0.08
Location, Site, Culture	0.05
Ecofact	0.3
Time	0.01
Pubauthor, Publoc, Pubyear	0
Feature, Person, Puborg	-

Tabelle 5: Bound[ary] Id[entification] Error out of the total of NE (both B- and I-) per category

respectively. These gazetteers were used to compute two binary valued features included into CRF. The features fire iff the current token matches with any element of the gazetteers. The system is retrained by including this feature to the previous feature set (c.f. Section 4.2) and keeping all other parameters unaltered. Overall evaluation results with two different thresholds 0.2 and 0.1 are reported in Table 4. We here again experimented with the selection criteria that not only adds the current sentence but also adds the surrounding sentences (preceding one and following one sentences). We have also shown in Table 4 that the number of sentences and number of named entities added from the development set to the training set in each iteration. At the end of 10th iteration with threshold equals to 0.1, it shows the overall recall, precision and F-measure values of 69.19%, 71.19%, and 70.18%, respectively. Again at the end of 10th iteration with threshold equals to 0.2, it shows the overall recall, precision and F-measure values of 69.51%, 71.47%, and 70.48%, respectively. Comparisons between Table 3 and Table 4 suggest that gazetteers help to improve the performance. The baseline model (based on random selection) showed the recall, precision and F-measure values of 68.66%, 70.51% and 69.57%, respectively. In table we also show the number of sentences and NEs that are added to the initial training data in each iteration. The instances of B- and I- are treated as two different counts for NEs.

4.4 Error analysis

We carried out two types of analysis: of the ability of the system to identify named entity boundaries (here called **identification problem**), and of its ability to correctly classify the mentions (**classification problem**).

To evaluate identification, we calculated the amount of mismatches between B-subtype and I-subtype for every class: those cases in which the system succeeds in recognizing the NE class, but fails to identify the correct bound. We only considered correctly identified entities (e.g. a true positive Artefact), calculating, among these, the error rate due to border mismatches (e.g. a B-Artefact marked as I-Artefact or viceversa).

In Table 5 we report the boundary identification error out of the total amount of NE per class.[4] In most cases, the problem of border identification lies in the ability of the system of incorporating the complex preposition which opens the mention; the lack of a

[4] Given the classes B-Artefact and I-Artefact, we calculated the ratio between the *FN*s and the population(B-artefact+I-artefact). Since we consider only cases in which the entity is correctly identified, we end up having a binary situation (either *b-entity* or *i-entity*); thus, FPs are not relevant as they overlap with the FNs of the other class.

Class	TP	FP	FN	Tot Retr	Total	P	R	F-M
B-Artefact	26	70	21	96	47	0.27	0.55	0.36
B-Culture	12	34	17	46	29	0.26	0.41	0.32
B-Ecofact	164	37	107	201	271	0.82	0.61	0.69
B-Feature	0	9	0	9	-	0	-	
B-Location	117	78	52	195	169	0.6	0.69	0.64
B-Person	0	20	0	20	-	0	-	
B-Pubauthor	380	23	55	403	435	0.94	0.87	0.91
B-Publoc	2	1	3	3	5	0.67	0.4	0.5
B-Puborg	1	0	7	1	8	1	0.13	0.22
B-Pubyear	265	20	10	285	275	0.93	0.96	0.95
B-Site	57	64	66	121	123	0.47	0.46	0.47
B-Time	97	14	44	111	141	0.87	0.69	0.77
I-Artefact	70	76	27	146	97	0.48	0.72	0.58
I-Culture	20	48	26	68	46	0.29	0.43	0.35
I-Ecofact	232	40	121	272	353	0.85	0.66	0.74
I-Feature	0	0	14	0	14	-	0	-
I-Location	262	164	66	426	328	0.62	0.8	0.69
I-Person	0	0	24	0	24	-	0	-
I-Pubauthor	64	9	40	73	104	0.88	0.62	0.72
I-Publoc	6	0	30	6	36	1	0.17	0.29
I-Puborg	13	1	24	14	37	0.93	0.35	0.51
I-Pubyear	0	0	2	0	2	-	0	-
I-Site	168	98	95	266	263	0.63	0.64	0.64
I-Time	400	40	66	440	466	0.91	0.86	0.88
Total	2356	817	946	3173	3302	-	-	-
O	11703	126	38	11829	11741	0.99	1	0.99

Tabelle 6: Precision and Recall per class

consistent number of these mentions in the training set can be behind this difficulty.

Classification accuracy is a measure of the system w.r.t. its ability to correctly assign the exact class to the identified NE. As shown in Table 6, there are categories in which the NE tagger obtains very good results, such as Pub-year, Pub-author, and Time. (Not surprisingly, these are among the classes most frequently studied in HLT.) On the other hand, categories such as Artefact, Culture and Site are more difficult to classify. These classes are also difficult for coders, which suggests that in part the problem may be that this new domain still isn't well understood.

More specifically, the most frequent confusions are a) Culture vs Site and vs Time b) Site vs Location and c) Ecofact vs Artefact. The confusions under a) were expected, because the classes Culture and Site, and Culture and Time, are systematically correlated: e.g., many cultures such as *Starcevo* are so-named from a so-called **type site**. As a result, whereas 55% of Culture NEs are correctly identified, 20% are marked as Site. To study this issue we asked annotators to mark mentions they felt could instantiate to different classes with two labels, *label 1* (the more likely) and *label 2*, and set an underspecification attribute (see (Poesio and Artstein, 2005) for a more extensive study of this type of annotation), and we found that the cases of confusion had often been marked as underspecified.[5] In class b), Site vs Location, 70% of Site NEs

[5] Though human annotators mark these entities with two labels, during training, the NE tagger choses only the first one, the one considered most likely; for this reason the underspecification issue does not affect the evaluation phase.

are correctly identified, but 14% is marked as `Location`. In this case we have a semantic ambiguity between classes that share similar context: e.g. *nella vicina Alta Valtrompia* vs *il sito nei pressi di Bressanone*. As expected, the introduction of the Gazetteer reduced the distance in particular in this case. Finally, for class c), `Ecofact- Artefact`, 65,5% of `Ecofact` NE a are correctly identified, but 19% are marked as `Artefact`, while only 5% is confused with `Location`, which is the second most confused class. This case also concerns a critical distinction, often marked as `underspecified` by our coders, and the focus of ongoing discussions in the domain experts community.[6]

5 Conclusions

Our results suggest, first of all, that active annotation does lead to better results than random sampling; and second, that our approach leads to reasonable results with relatively small amounts of trained data. Our future work will include, first of all, revising the coding scheme for the Archaeology domain in collaboration with the Archaeology experts, in particular developing a solution to the Underspecification problem and carrying out agreement tests; and testing the generality of our results by incorporating a new domain.

Acknowledgments

This work was in part supported by the LiveMemories project.

Literatur

Briscoe, E. e. a. (2011). Intelligent information access from scientific papers. In et al, J. T., editor, *Current Challenges in Patent Information Retrieval*. Springer.

Buitelaar, P. (1998). *CoreLex : Systematic Polysemy and Underspecification*. PhD thesis, Brandeis University.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289.

Pianta, E., Girardi, C., and Zanoli, R. (2008). The textpro tool suite. In *Proc. of 6th LREC*, Marrakech.

Poesio, M. and Artstein, R. (2005). The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In Meyers, A., editor, *Proc. of ACL Workshop on Frontiers in Corpus Annotation*, pages 76–83.

Poesio, M., Barbu, E., Bonin, F., Cavulli, F., Ekbal, A., Saha, S., Stemle, E., Girardi, C., Nardis, D., and Nardelli, F. (2011a). The humanities research portal: Human language technology meets humanities publication repositories. In *Proc. of SDH*, Copenhagen.

Poesio, M., Barbu, E., Stemle, E., and Girardi, C. (2011b). Structure-preserving pipelines for digital libraries. In *Proc. of LaTeCH*, Portland, OR.

[6] In this domain there is also a systematic ambiguity between `Culture` and `Artefact`, because of the tradition to name other cultures according to their most distinctive artefact, as in: *Cultura dei Vasi a bocca quadrata.*

Settles, B. (2009). Active learning literature survey. In *In Computer Sciences Technical Report 1648 University of Wisconsin-Madison*.

Sha, F. and Pereira, F. (2003). Shallow Parsing with Conditional Random Fields. In *Proceedings of NAACL '03*, pages 134–141, Canada.

Vlachos, A. (2006). Active annotation. In *Proc. EACL 2006 Workshop on Adaptive Text Extraction and Mining*, Trento.