Sentence-Based Active Learning Strategies for Information Extraction

Andrea Esuli, Diego Marcheggiani* and Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi, 1 – 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

ABSTRACT

Given a classifier trained on relatively few training examples, *active learning* (AL) consists in ranking a set of unlabeled examples in terms of how informative they would be, if manually labeled, for retraining a (hopefully) better classifier. An important text learning task in which AL is potentially useful is *information extraction* (IE), namely, the task of identifying within a text the expressions that instantiate a given concept. We contend that, unlike in other text learning tasks, IE is unique in that it does not make sense to rank individual items (i.e., word occurrences) for annotation, and that the minimal unit of text that is presented to the annotator should be an entire sentence. In this paper we propose a range of active learning strategies for IE that are based on ranking individual sentences, and experimentally compare them on a standard dataset for named entity extraction.

Keywords

Information extraction, named entity recognition, active learning, selective sampling

1. INTRODUCTION

In many applicative contexts involving supervised learning, labeled data may be scarce or expensive to obtain, while unlabeled data, even sampled from the same distribution, may abound. In such situations it may be useful to employ an algorithm that ranks the unlabeled examples and asks a human annotator to label a few of them, starting from the top-ranked ones, so as to provide additional *highly informative* training data. The task of this algorithm is thus to rank the unlabeled examples in terms of how informative they would be, once labeled, for the supervised learning task. The discipline that studies these algorithms is called *(pool-based) active learning* (aka *selective sampling*). This paper focuses on the application of active learning to *information extraction* (IE), the task of annotating sequences of one or more words (aka *tokens*) in a text by means of *tags* representing concepts of interest. The hypothetically perfect IE system is thus the one for which, for each tag in the *tagset* of interest, the *predicted sequences* of tokens coincide with the *true sequences*.

In text classification and other text learning tasks different from IE, the units of ranking and the units of annotation are the same; e.g., in text classification, it is the texts themselves that are ranked, and it is the texts themselves that are then annotated in their entirety by the human annotator. IE is peculiar from this standpoint since, while the units of annotation are the tokens, it does not make sense to *rank* individual tokens: if this were to happen, an annotator would be presented with "tokens in context" (i.e., a token in the fixed-size window of text in which the token occurs) and asked to annotate the token, with the consequence that she might be asked to read the same context several times, for annotating neighbouring tokens.

In this paper we take the view that the optimal unit of ranking is the *sentence*. This means that all the sentences of the automatically annotated texts are going to be ranked and presented to the annotator, who will then annotate *all* the tokens of a few sentences, starting from the top-ranked ones. This is different from several other works in the field [6, 8, 12], in which the unit of ranking is a portion of text smaller than a sentence, i.e., a predicted sequence embedded in a fixed-sized text window a few words long. The problem with the latter approach is that, by focusing on *predicted* sequences, the classification mistakes that the annotator corrects are the false positives, while the false negatives are never brought to the light. This results in an imbalanced training set being fed to the learner.

We deem the sentence to be the optimal unit of ranking for additional reasons:

- An entire sentence offers more context for actually interpreting the tokens and the sequences within it than the fixed-size window often used in the literature. This is especially important in complex IE tasks such as opinion extraction (see e.g., [2, 5]), in which, given the variety of devices that language has for conveying opinions, and given the uncertain boundary between fact and opinion, the annotator needs to take very subtle decisions.

- Different sentences never overlap, while different fixed-length windows may do. The sentence-based approach results in smaller annotation effort, since the same token is never examined twice by the annotator.

- From a semantic point of view, sentences are fairly self-contained units. This means that using portions of

*Corresponding author

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

text *larger* than sentences (e.g., paragraphs) as ranking units is unnecessary, also given that it is hardly the case that an annotation crosses the boundary between two consecutive sentences. Conversely, with a fixed-size window centered around a predicted sequence, another true sequence may cross the boundary between the window and its neighbouring text.

In the past, typical strategies adopted in AL for generic learning tasks have relied on ranking objects based either on the classification score attributed by the classifier to the object (*relevance sampling*), or on the confidence score with which the classifier has classified it (*uncertainty sampling*) [9]. In IE, if we want to rank entire sentences we have to come to terms with the fact that *each* token in the sentence has obtained a classification and a confidence score for *each* tag in the previous classification round, and we thus have to generate a sentence-specific score out of the token- and tag-specific scores, for all the tokens contained in the sentence and all the tags in the tagset.

The main contribution of this paper consists in proposing several alternative strategies for combining the token- and tag-specific scores into a sentence-specific score, and comparing these strategies experimentally.

We remark that this paper does *not* deal with active learning algorithms for specific supervised learning devices (such as e.g., [13] for text classification), but presents active learning strategies that are independent of the learning device and that are thus in principle suitable for use with any such device.

The rest of the paper is organized as follows. Our strategies for performing AL in IE are described in Section 2. In Section 3 we move to describing our experiments and the experimental protocol we have followed. We conclude in Section 4 by pointing out avenues for future work.

2. ACTIVE LEARNING STRATEGIES FOR INFORMATION EXTRACTION

2.1 Preliminaries: Information Extraction

This paper focuses on the application of active learning to (*single-tag*) *information extraction* (STIE, or simply IE). Let a text T consist of a sequence $T = \{t_1 \prec s_1 \prec \ldots \prec s_{n-1} \prec t_n\}$ of *tokens* (i.e., word occurrences) and *separators* (i.e., sequences of blanks and punctuation symbols), where "$\prec$" means "precedes in the text". Let $C = \{c_1, \ldots, c_m\}$ be a predefined set of *tags* (aka *labels*, or *classes*), and let $c_\emptyset \notin C$ be a special tag (to be read as "no tag"). We define (single-tag) information extraction as the task of estimating an unknown *target function* $\Phi : T \rightarrow C \cup \{c_\emptyset\}$ that specifies the true tag in $C \cup \{c_\emptyset\}$ attached to each token $t_i \in T$ and to each separator $s_i \in T$. The result $\hat{\Phi} : T \rightarrow C \cup \{c_\emptyset\}$ of this estimation is called the *tagger* (or *wrapper*, or *classifier*)[1]. A further property of both Φ and $\hat{\Phi}$ is that they can attribute a tag c_j to a separator s_i only if they also attribute the same tag to both t_{i-1} and t_i.

In most IE tasks it is actually the case that, rather than isolated tokens and separators, *sequences* of consecutive tokens and separators are annotated with a given tag; e.g., the sequence "George W. Bush", containing three tokens and two separators, might be annotated with the PER ("person name") tag. Such sequences of tokens will here be referred to as *annotated sequences* (ASs); the expressions *true AS* and *predicted AS* will refer to ASs according to Φ and $\hat{\Phi}$, respectively. Note that the reason for considering separators to be the object of tagging too is that the IE system should correctly identify sequence boundaries. For instance, given the expression "Barack Obama, Hillary Clinton and Joe Biden" the perfect IE system will attribute the PER tag, among others, to the tokens "Barack", "Obama", "Hillary", "Clinton", and to the separators (in this case: blank spaces) between "Barack" and "Obama" and between "Hillary" and "Clinton", but *not* to the separator ", " between "Obama" and "Hillary". If the IE system does so, this means that it has correctly identified the boundaries of the sequences "Barack Obama" and "Hillary Clinton".

Note that "single-tag" IE means that each token (resp., separator) has exactly one tag. This is different from multi-tag IE, in which it is assumed that a given token (resp., separator) may have more than one tag (opinion extraction – see e.g., [5] – is an instance of multi-tag IE).

2.2 Sentence-Based AL strategies for IE

Our experimental work is focused on comparing a range of active learning strategies for IE that are based on ranking individual sentences. This section describes the strategies and the intuitions supporting them.

In this work we test two alternative learning devices, *support vector machines* (SVMs) (see e.g., [1]), and *conditional random fields* (CRFs) [7]. For SVMs we have adopted a widely used method to realize a multiclass classifier as a combination of binary classifiers, i.e., a *one versus all* method. The one versus all method consists in learning m binary classifiers $\hat{\Phi}_c : T \rightarrow \mathbb{R}$, each one trained using as the positive examples all the tokens in the training set Tr that are labeled with c, and as negative examples all the other tokens, regardless of the original label. The multiclass classifier is then defined as $\hat{\Phi}(t) = \arg\max_{c \in C \cup \{c_\emptyset\}} \hat{\Phi}_c(t)$, i.e., the assigned label is the one whose binary classifier scored the maximum confidence.

CRFs are a discriminative probabilistic learning method based on an undirected graph model, and is frequently used for labeling sequential data, e.g., a sequence of words composing a text. Given a token t, a CRFs classifier estimates the likelihood $\hat{\Phi}_c(t) = P(c|t)$ for each $c \in C \cup \{c_\emptyset\}$ and, similarly to SVMs, the assigned label is the one scoring the highest $\hat{\Phi}_c(t)$ value. CRFs are nowadays considered the state-of-the-art learning device for information extraction [11].

The strategies we propose are based on two concepts, *label score* and *tag score*. The label score of a token is equal to $ls(t) = \max_{c \in C \cup \{c_\emptyset\}} \hat{\Phi}_c(t)$, i.e., the maximum confidence score that determines the decision taken by the classifier $\hat{\Phi}(t)$. The tag score is instead defined as $ts(t) = \max_{\{c \in C\}} \hat{\Phi}_c(t)$, i.e., the maximum confidence that the classifier as on considering a token as belonging to a tag, regardless of the confidence with respect to $c_\emptyset$.

2.2.1 Tag score-based strategies

The following strategies are based on combining the label scores assigned to the tokens in the sentence, following the intuition that the elements on which the classifier has low confidence could be more useful to the learner, so as to gather knowledge on "difficult" cases.

[1]Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

The *Min Min Confidence* (MMC) strategy assigns to the sentence a value equal to the minimum tag score value among the tokens composing it, i.e.,

$$MMC(s) = min_{t \in s}(ts(t)) \quad (1)$$

Sentence ranking is performed in increasing order of $MMC(s)$ value.

Min Average Confidence (MAC) is a version of MMC that tries instead to be robust to single "extreme" evaluations, averaging the tag scores of all the tokens composing the sentence, i.e.,

$$MAC(s) = avg_{t \in s}(ts(t)) \quad (2)$$

2.2.2 Label score-based strategies

Symmetrically to the tag-score-based strategies, the label-score-based strategies follow the somehow different intuition that the elements on which the classifier has high confidence could be useful, so that the strong beliefs of the learner are confirmed when correct or corrected when a blatant error is found.

The *Max Max Score* (MMS) strategy assigns to each sentence a value equal to the highest label score among the tokens composing it, i.e.,

$$MMS(s) = max_{t \in s}(ls(t)) \quad (3)$$

Sentence ranking is performed in decreasing order of $MAS(s)$.

Similarly to MAC, *Max Average Score* (MAS) instead averages the label scores of all the tokens composing the sentence, i.e.,

$$MAS(s) = avg_{t \in s}(ls(t)) \quad (4)$$

2.2.3 Tag count-based strategies

The following strategies are instead based on counting the number of tokens that are given a tag different from $c_\emptyset$ by the classifier.

The *Max Tag Count* (MTC) strategy counts the number of tokens in the sentence that are given a tag different from $c_\emptyset$, i.e.,

$$MTC(s) = |\{t \in s | \hat{\Phi}(t) \in C\}| \quad (5)$$

Sentence ranking is performed in decreasing order of $MTC(s)$ value.

Since MTC naturally favours long sentences, we have also tested a strategy (*Max Tag Ratio* – MTR) that normalizes the values by sentence length, i.e.,

$$MTR(s) = \frac{|\{t \in s | \hat{\Phi}(t) \in C\}|}{|s|} \quad (6)$$

The *Medium Tag Ratio* (MEDTR) strategy instead top-ranks the sentences with a tag ratio closer to the average tag ratio measured on the training set, i.e.,

$$MedTR(s) = \frac{MTR(s)}{avg_{s' \in Tr} MTR(s')} \quad (7)$$

2.2.4 Round Robin-based strategies

While the previous strategies always combine the confidence values returned on the various tag types, the following strategies are based on computing values separately for each tag, then selecting the most informative sentences using a "round robin" selection process across all the tags.

The *Round Robin Max Score* (RRMS) strategy assigns, for each $c \in C$, a relevance score to the sentence equal to the maximum score obtained by the tokens contained in it, i.e.,

$$RRMS_c(s) = max_{t \in s}(\hat{\Phi}_c(t)) \quad (8)$$

Then a round robin selection process is performed on the $|C|$ rankings produced.

Similarly to MAS, *Round Robin Average Score* (RRAS) uses averaging instead of maximization, i.e.,

$$RRAS_c(s) = avg_{t \in s}(\hat{\Phi}_c(t)) \quad (9)$$

The *Round Robin Max Tag Ratio* (RRMTR) strategy applies instead the MTR strategy considering the various tags separately from each other, so as to avoid favouring the most frequent tags over the most infrequent.

3. EXPERIMENTS

3.1 Experimental setting

The dataset we have used for evaluating our strategies is the CoNLL2003 named entity extraction dataset. The dataset consists of 1,393 Reuters newswire articles, for a total of 301,418 tokens. The tagset consists of 4 tags (LOC, PER, ORG, MISC, standing for "location", "person", "organization", and "miscellaneous", respectively) plus the special tag O, which tags any token / separator not tagged by any tag in {LOC, PER, ORG, MISC}. The tokens inside the corpus are tagged as follows: 10,645 tokens are tagged as LOC, 9,323 as ORG, 10,059 as PER, 5,062 as MISC, while the remaining 266,329 are tagged as O. We used a version of the CoNLL corpus already preprocessed with Pianta and Zanoli's Tagpro system [10], a PoS-tagging system based on YamCha that computes features such as prefixes, suffixes, orthographic information (e.g., capitalization, hyphenation) and morphological features, as well as PoS tags and chunk tags. These features altogether form the vectorial representations of tokens and separators that are fed to the learning device.

For this latter, we have tested two alternative, off-the-shelf packages, i.e., YamCha[2] and CRF++[3], respectively based on support vector machines and conditional random fields.

We evaluate the results of our experiments using the F_1 measure on a *token & separator* evaluation model [3]. The token & separator model considers each token and each separator as being the objects of tagging; for instance, given tag c, the TP ("true positives") entry of the contingency table for c consists in the number of tokens that are correctly assigned token c plus the number of separators that are correctly assigned token c. Once the contingency tables for all the tags in C have been filled, the evaluation is done by using standard micro-averaged and macro-averaged F_1.

3.2 Experimental protocol

In this work we adopt the following iterative experimental protocol. The protocol has three integer parameters α, β, and γ. Let Ω be a set of natural language sentences partitioned into a training set Tr and a test set Te, and let σ be an active learning strategy:

1. Set an iteration counter $t = 0$;

[2] `http://www.chasen.org/~taku/software/YamCha/`
[3] `http://crfpp.sourceforge.net/`

2. Set the current training set Tr_t to the set of the first α sentences of Tr; set the current "unlabeled set" $U_t \leftarrow Tr/Tr_t$;
3. For $t = 1, \ldots, \beta$ repeat the following steps:
   (a) Generate a classifier $\hat{\Phi}^t$ from the current training set Tr_t;
   (b) Evaluate the effectiveness of $\hat{\Phi}^t$ on Te;
   (c) Classify U_t by means of $\hat{\Phi}^t$;
   (d) Rank U_t according to strategy σ, thus generating the ranking $\sigma(U_t)$;
   (e) Let $r(U_t, \gamma)$ be the smallest prefix of $\sigma(U_t)$ (i.e., the smallest number of top-ranked elements of $\sigma(U_t)$) that contains at least γ tokens; set $Tr_{t+1} \leftarrow Tr_t \cup r(U_t, \gamma)$; set $U_{t+1} \leftarrow U_t/r(U_t, \gamma)$.

It is important to remark that Step 3b has only the purpose of collecting the results for experimental purposes (i.e., for producing the tables of Section 3.3); since it uses the test set Te, its results should obviously not be (and are not) accessible to the algorithm.

The above protocol simulates the activity of a human annotator who, at the beginning of the process, has available a training set Tr_0 consisting of α manually tagged sentences, and an "unlabeled set" U_0 consisting of $|Tr| - \alpha$ untagged sentences. The annotator generates a classifier $\hat{\Phi}^0$ from Tr_0, uses it to tag the sentences in U_0, asks the active learning agent to rank them, manually labels the top-ranked ones for a total of roughly γ tokens, generates a new classifier $\hat{\Phi}^1$ from an augmented training set that comprises Tr_0 and the newly tagged sentences, and repeats this process β times.

In our experiments we have set $\alpha = 110$ (in the CoNLL 2003 dataset this means approximately 2000 tokens), $\beta = 20$, and $\gamma = 200$; this means that each strategy will be evaluated by testing the accuracy of the classifiers generated from training sets consisting of approximately 2000, 2200, ..., 5800, 6000 training tokens, for a total 20 experiments per strategy. We think these parameters are realistic, since they simulate a situation in which

- there are only about 100 manually tagged sentences at the beginning; (this is reasonable, since in many applications in which significantly more training data are available, human annotators might not find it worthwhile to annotate any further);
- every time the human annotator manually labels 200 unlabeled tokens, he/she wants to retrain the system; (this is reasonable, since he/she wants the operate on a ranking of the unlabeled documents that incorporates as much as possible the feedback he/she has already given to the system;)
- the human annotator does not want to do any further manual labeling once about 6,000 training tokens are available; (this seems reasonable, since at this point the cost-effectiveness of the manual effort has probably decreased significantly.)

As the baseline strategy for the evaluation of our results we adopt the one that consists in adding further labeled sentences to the training set by picking them at random. This simulates the behaviour of a human annotator that picks unlabeled sentences and labels them in no particular order.

3.3 Results

The main results of our experiments are summarized in Table 1. This table reports, for each individual strategy, the values of F_1^μ and F_1^M obtained after 20 training sessions resulting from the protocol of Section 3.2, with $\alpha = 110$, $\beta = 20$, and $\gamma = 200$, using the two different learners, SVMs and CRFs.

Quite surprisingly, the only genuine strategy that outperforms the random baseline is the MAC strategy. The relative improvement of MAC over RAND ranges from 3.9% up to 6.3%. This improvement matches our expectations, given the close relation between the MAC strategy with the *uncertainty sampling* [9] method which already proved to be effective for AL.

Surprisingly, all the other strategies perform worse or no better than the random baseline. In order to understand the possible motivations behind these results we have inspected the sentences selected by the various strategies at the various iterations. This inspection allowed us to draw some specific conclusions on some of the strategies, and a general observation for the entire pool of strategies.

The MTR and RRMTR strategies tend to select very short sentences (two/three words) composed just by named entities. This allows gathering a lot of different instances of named entities, but without a context of use, which is important in order to learn how to perform extraction from longer, more articulated sentences.

The MTC strategy selects sentences of variable length, but tends to exceed in selecting sentences full of named entities, thus with a very limited amount of O-tagged tokens.

A common aspect of all the strategies is that, the more similar two sentences are, the more similar are the scores that the various strategies assign them. If the dataset contains a lot of similar sentences, and such sentences obtain high scores, the contribution of relevant information to the training set is limited, because of the redundancy contained in the set of sentences selected.

A comparison between the strategies based on round robin (RRAS, RRMS, RRMTR) against the respective "single-rank" versions (MAS, MMS, MTR) shows that the RR-strategies produce an improvement in the F_1^M measure, as should be expected when using a class-balancing method as RR.

The comparison of the averaging-based strategies (MAC, MAS, RRAS) against the respective versions based on maximization / minimization (MMC, MMS, RRMS) shows that averaging always perform better than maximization / minimization. This indicates that the smoothing introduced by the averaging helps the strategies to filter out the single "false-relevant" tokens that may appear in otherwise non-relevant sentences.

4. CONCLUSIONS

We have argued that, in active learning for information extraction, the sentence should be the unit of ranking. We have thus studied several strategies for scoring a given sentence for ranking, based on the classification score and the confidence score obtained by each token in the sentence. On the positive side, the experimental results that we have obtained by testing these strategies on a named entity extraction task show one such strategy (Min Average Confidence) to outperfom the others, irrespectively of learning device

		base	MAC	MAS	MMC	MMS	RRAS	RRMS	MTR	RRMTR	MTC	MedTR
F_1^μ	YamCha	.650	**.683**	.583	.645	.530	.639	.596	.628	.607	.632	.555
	CRF++	.656	**.697**	.610	.654	.463	.643	.573	.622	.509	.626	.544
F_1^M	YamCha	.634	**.661**	.525	.633	.526	.644	.577	.593	.597	.613	.538
	CRF++	.639	**.664**	.546	.634	.473	.633	.564	.563	.519	.606	.517

Table 1: Values of F_1 obtained after the last training session, i.e., with classifiers trained on approximately 2,000 training tokens plus approximately 4,000 tokens manually annotated as a result of the active learning strategy. Boldface indicates the best performance.

used (support vector machines or conditional random fields) and evaluation measure (microaveraged or macroaveraged F_1) used. On the negative side, the same results show that all the other strategies, that seem based on solid intuitions, tend to be roughly equivalent to a random strategy. In the future we plan to test these strategies further, possibly on IE tasks more difficult than named entity extraction such as opinion extraction.

Acknowledgments

We thank Emanuele Pianta and Roberto Zanoli for kindly providing us a version of the CoNLL corpus already preprocessed with their Tagpro system [10].

5. REFERENCES

[1] Christopher J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, Sydney, AU, 2006.

[3] Andrea Esuli, Michał Pryczek, and Fabrizio Sebastiani. Evaluating information extraction systems. Technical report, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 2010. Forthcoming.

[4] Andrea Esuli and Fabrizio Sebastiani. Active learning strategies for multi-label text classification. In *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, pages 102–113, Toulouse, FR, 2009.

[5] Andrea Esuli and Fabrizio Sebastiani. Enhancing opinion extraction by automatically annotated lexical resources. In *Proceedings of the 4th Language Technology Conference (LTC'09)*, pages 224–228, Poznań, PL, 2009.

[6] Rosie Jones, Rayid Ghani, Tom Mitchell, and Ellen Riloff. Active learning for information extraction with multiple view feature sets. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM'03)*, number 18–25, Cavtat–Dubrovnik, KR, 2003.

[7] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 282–289, Williamstown, US, 2001.

[8] Florian Laws and Hinrich Schütze. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 465–472, Manchester, UK, 2008.

[9] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 3–12, Dublin, IE, 1994.

[10] Emanuele Pianta and Roberto Zanoli. Tagpro: A system for Italian POS tagging based on SVMs. *Intelligenza Artificiale*, 4(2):8–9, 2007.

[11] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261—377, 2008.

[12] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 589–596, Barcelona, ES, 2004.

[13] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.