Information Extraction with Active Learning: a Case Study in Legal Text

Cristian Cardellino[1], Serena Villata[2], Laura Alonso Alemany[1], and Elena Cabrio[2]

[1] Universidad Nacional de Córdoba, Argentina
crscardellino@gmail.com, alemany@famaf.unc.edu.ar
[2] INRIA Sophia Antipolis, France
firstname.lastname@inria.fr

Abstract. Active learning has been successfully applied to a number of NLP tasks. In this paper, we present a study on Information Extraction for natural language licenses that need to be translated to RDF. The final purpose of our work is to automatically extract from a natural language document specifying a certain license a machine-readable description of the terms of use and reuse identified in such license. This task presents some peculiarities that make it specially interesting to study: highly repetitive text, few annotated or unannotated examples available, and very fine precision needed.
In this paper we compare different active learning settings for this particular application. We show that the most straightforward approach to instance selection, uncertainty sampling, does not provide a good performance in this setting, performing even worse than passive learning. Density-based methods are the usual alternative to uncertainty sampling, in contexts with very few labelled instances. We show that we can obtain a similar effect to that of density-based methods using uncertainty sampling, by just reversing the ranking criterion, and choosing the *most certain* instead of the *most uncertain* instances.

Key words: Active Learning, Ontology-based Information Extraction

1 Introduction and Motivation

Licenses and data rights are becoming a crucial issue in the Linked (Open) Data scenario, where information about the use and reuse of the data published on the Web need to be specified and associated to the data. In this context, the legal texts describing the licenses need to be translated into machine-readable ones to allow for automated processing, verification, etc.

Such machine-readable formulation of the licenses requires a high degree of reliability. For example, if the original license states that action A is forbidden and this prohibition is not reported in the RDF version of the license then this could lead to misuses of the data associated to that machine-readable license. For this reason, we need highly accurate performance in the task, to guarantee highly reliable outputs.

In this scenario, human intervention is unavoidable, to establish or validate the correspondence between concepts in ontologies and expressions in natural language. In this paper, we propose to ease this dependency by optimizing human intervention through an active learning approach. Active learning techniques [13] aim to get powerful insights on the inner workings of automated classifiers and resort to human experts to analyze examples that will most improve their performance. We show the boost in performances introduced by different improvements on a classical, machine learning approach to information extraction.

More precisely, in the experimental evaluation of our framework, we show that active learning produces the best learning curve, reaching the final performance of the system with fewer annotated examples than passive learning. However, the standard active learning setting does not provide an improvement in our study case, where very few examples are available. Indeed, if we choose to annotate first those instances where the classifier shows more uncertainty, the performance of the system does not improve quickly, and, in some cases, it improves more slowly than if instances are added at random. In contrast, selecting for annotation those instances where the classifier is most certain (*reversed uncertainty sampling*) does provide a clear improvement over the passive learning approach. It is well-known that uncertainty sampling does not work well with skewed distributions or with few examples, in those cases, density estimation methods work best. We show that using *reversed uncertainty sampling* in this particular context yields results in the lines of density estimation methods.

The rest of the paper is organized as follows: in Section 2 we discuss the general features of the active learning approach and related work, Section 3 presents our approach to ontology-based IE for licenses; in Section 4 we describe how we apply active learning techniques to this kind of problems. Experimental results comparing the different approaches are discussed in Section 5.

2 Relevant Work

Active learning [13] is a more "intelligent" approach to machine learning, whose objective is to optimize the learning process. This optimization is obtained by choosing examples to be manually labelled, by following some given metric or indicator to maximize the performance of a machine learning algorithm, instead of choosing them randomly from a sample. This capability is specially valuable in the context of knowledge-intensive Information Extraction, where very obtaining examples is costly and therefore optimizing examples becomes crucial.

The process works as follows: the algorithm inspects a set of unlabeled examples, and ranks them by how much they could improve the algorithm's performance if they were labelled. Then, a human annotator (the so-called "oracle") annotates the highest ranking examples, which are then added to the starting set of training examples from which the algorithm infers its classification model, and the loop begins again. In some active learning approaches, the oracle may annotate features describing instances, and not (only) instances themselves. This latter approach provides even faster learning in some cases [6, 12, 10, 15].

Different strategies have been applied to determine the most useful instances to be annotated by the oracle, including expected model change, expected error reduction or density-weighted methods [11]. The most intuitive and popular strategy is *uncertainty sampling* [9], which chooses those instances or features where the algorithm is most uncertain. This strategy has been successfully applied to Information Extraction tasks [3, 14]. Uncertainty can be calculated by different methods depending on the learning algorithm. The simplest methods exploit directly the certainty that the classifier provides for each instance that is classified automatically. This is the information that we are exploiting.

However, we did not only use uncertainty sampling, but also the exact opposite. We explored both prioritizing items with highest certainty and with lowest certainty. We followed the intuition that, when a model is very small, based on very few data, it can be improved faster by providing evidence that consolidates the core of the model. This is achieved by choosing items with highest certainty, because they also provide the lowest entropy with respect to the model, and can help to redirect wrong assumptions that a model with very few data can easily make. When the core of the model is consolidated, items with highest uncertainty should provide a higher improvement in performance by effectively delimiting with more precision the decision frontier of the model. This phenomenon, which lies at the heart of well-known semi-supervised learning techniques like self-training (or *bootstrapping*), has also been noted by approaches combining density estimation methods when very few examples are available, and uncertainty sampling when the training dataset has grown [5, 17].

Other approaches have been applied to fight the problem of learning with few examples, by finding the optimal seed examples to build a training set [4, 7]. However, these approaches are complex and difficult to implement, thus lie beyond the capacities of the regular NLP practitioner. In contrast, the approach presented here is conceptually simple and easy to implement, as it is a wrapper method over your best-know classifier.

We developed an active learning tool inspired on Dualist [12]. As in Dualist, we provide a graphical user interface for the human oracle to answer the queries of the active learning algorithm. The base machine learning algorithm is also a Multinomial Naïve Bayes, but our method for ranking instances is uncertainty/certainty sampling based on the confidence of the classifier. Features can also be labelled, using Information Gain to select them, but sequentially with respect to instances, not simultaneously as in Dualist. As an addition, our approach allows for multiclass labeling, that is, an instance can be labelled with more than one class. Our active learning framework source together with the dataset is available at `https://github.com/crscardellino/nll2rdf-active-learner`.

3 Passive learning IE system for textual licenses

As a base to our system, we used NLL2RDF, an Information Extraction system for licenses expressed in English, based on a (passive) machine learning approach [1]. The final goal of the system is to identify fragments of text that

allow to identify a *prohibition*, a *permission* or an *obligation* (or *duty*) expressed by a license. When these fragments are identified, they are converted into an RDF machine-readable specification of the license itself. Section 3.1 provides a general overview of the system describing the representation of licensing information we selected, and Section 3.2 presents the machine learning approach adopted whithin the system, as well as the performances of the basic setting.

3.1 Overview of the system

The architecture of the system is based on a machine learning core, with an SVM classifier that learns from examples. Examples are manually assigned to one of a predefined set of classes associated to the licenses ontology. Many vocabularies exist to model licensing information. Some examples include LiMO[3], L4LOD[4], ODRS[5] and the well known Creative Commons Rights Expression Language (CC REL) Ontology[6]. So far the Linked Data community has mainly used the CC REL vocabulary, the standard recommended by Creative Commons, for machine-readable expression of licensing terms.

However, more complex licenses information can be defined using the Open Digital Rights Language (ODRL) Ontology[7], that allows to declare rights and permissions using the terms as defined in the Rights Data Dictionary[8]. This vocabulary, in particular, has not been specifically conceived for the Web of Data scenario, but it intends to provide flexible mechanisms to support transparent and innovative use of digital content in publishing, distribution and consumption of digital media across all sectors. ODRL allows to specify fine grained licensing terms both for data (thus satisfying the Web of Data scenario) and for all other digital media. The ODRL vocabulary defines the classes to which each text fragment needs to be translated by the system. It specifies different kinds of Policies (i.e., Agreement, Offer, Privacy, Request, Set and Ticket). We adopt `Set`, a policy expression that consists in entities from the complete model. Permissions, prohibitions and duties (i.e., the requirements specified in CC REL) are specified in terms of an `action`. For instance, we may have the action of attributing an `asset` (anything which can be subject to a policy), i.e., `odrl: action odrl: attribute`. For more details about the ODRL vocabulary, refer to the ODRL Community group.[9]

3.2 Machine learning core

[3] `http://data.opendataday.it/LiMo`
[4] `http://ns.inria.fr/l4lod/`
[5] `http://schema.theodi.org/odrs/`
[6] `http://creativecommons.org/ns`
[7] `http://www.w3.org/ns/odrl/2/`
[8] `http://www.w3.org/community/odrl/`
[9] `http://www.w3.org/community/odrl/`

The core of the system is based on passive machine learning. Given some manually annotated instances, a classifier is trained to assign each text fragment to one or more of the given ontological classes, including the class of instances that is not associated to any meaning in the reference ontology (i.e., ODRL in this case), which is the case for the majority of sentences in any given license.

In the first approach, a Support Vector Machine classifier was used. Texts were characterized by the unigrams, bigrams and trigrams of lemmas, obtaining an f-measure that ranged from 0.3 to 0.78 depending on the class, with 0.5 average. Later on we included bigrams and trigrams of words that co-occur in a window of three to five words. This last feature is aimed to capture slight variations in form that convey essentially the same meaning.

These additional features increased the average accuracy of the system to 76%, kappa coefficient of .7. Although the performance of the system was fairly acceptable in general, it was not acceptable considering that we are dealing with legal information, and that an error in the system could cause an actual misuse of the data. Moreover, we found that it was difficult to improve such performances given the complexity of the task. Finally, we wanted to make it easier to port this system to other domains (i.e., other kind of legal documents like contracts, or policies), and to do that it was crucial to optimize the annotation effort (only 37 licenses where considered and annotated). For all these reasons, we decide to adopt an active learning setting.

In the active learning setting, we decide to use a different classifier that allowed easy manipulation of its inner workings, so that we could implement active learning tweaks easily. As in [12], a Multinomial Naïve Bayes (MNB) classifier was the classifier of choice.

As a baseline to assess the improvement provided by the active learning approach to the problem, we assess the performance of the MNB in a Passive Learning setting. The performance of the MNB by itself was quite below that of SVMs, of 63% (kappa coefficient of .6). Since it is well-known that bayesian methods are more sensitive to noise than SVMs, we applied Feature Selection techniques as a preprocessing to this classifier. We calculated the IG of each feature with respect to the classes, and kept only the 50 features with most IG, as long as they all had an IG over 0.001, those with IG below that threshold were discarded. Feature Selection yields an important improvement in performances, reaching an accuracy of 72%. This performance, however, is still below that of SVMs, and that is why we study a third improvement: one vs. all classification.

As pointed out above, MNB is highly sensitive to noise, which seems specially acute in this setting where we have only very few examples of many of the classes. To obtain better models in this context, we applied a one vs. all approach, where a different classifier is trained to distinguish each individual class from all the rest. This, combined with a separate Feature Selection preprocess for each of the classifiers yields a significant improvement in performances, reaching an accuracy of 83%, with a kappa coefficient of .8. This allows us to use MNB as a base classifier for active learning, without sacrificing loss in performance with respect to the SVM baseline. Results are summarized in Table 1.

	plain	with FS	one vs. all	one vs. all & FS	one vs. all & class-specific FS
SVM	76	76	71	73	73
MNB	63	72	60	78	**83**

Table 1. Accuracy of two passive learning classifiers with different configurations.

4 Licenses IE within an active learning loop

The benefits of active learning, as discussed before, are a faster learning curve and an optimization of the human effort needed to train an automatic classifier. We want to assess the impact of an active learning approach in the task of License Information Extraction.

We apply uncertainty sampling to assess the utility of instances and IG to assess the utility of features for a given model. We then explored the effects of ranking instances either by highest or lowest uncertainty.

We implemented a system to apply active learning to the kind of annotation that we aim to develop, with functionalities similar to those of Dualist [12]. The architecture of the system is visualized in Figure 1. The system is provided with an annotated and an unannotated dataset. A model is learnt from the annotated dataset, applying MNB in a one-vs-all setting with separated feature selection for each classifier. Then, the model is applied to an unannotated dataset, and instances in this dataset are ranked according to the certainty of the model to label them, ranking highest those with most certainty or with most uncertainty. The highest ranking instances are presented to the oracle, who annotates them, associating each instance to one or more of the classes defined by the ODRL ontology or the class "null" if none of the available classes apply for the instance.

Then the oracle is given the possibility to annotate features that she finds as clearly indicative of a given class. For each class, the list of features with highest IG with the class is provided, and the oracle selects those that she finds are indicative of the class. If the user chooses not to annotate features, they are selected by the automated feature selection technique, that is, the system keeps for each one-vs.-all classifier only the top 50 features with highest IG with the class or only those features with more than 0.001 IG with the class, whichever condition produces the biggest set. If the user chooses to annotate features, these are added to the pool of features selected with the default method.

Finally, the system is trained again with the annotated corpus, now enhanced with the newly annotated examples and possibly newly annotated features.

The system is built as a hybrid application with Perl[10], Scala[11] and Java[12]. It uses the libraries within Weka [16], including LibSVM [2], for the training and evaluation of the classifier and has a hybrid web application (that uses both Perl

[10] `https://www.perl.org/`
[11] `http://www.scala-lang.org/`
[12] `https://java.com/`

and Scala), to use in the local machine, created with `Play!`[13], a Model-View-Controller (MVC) Web application framework for the Scala language, in order to use HTML for a graphic interface.

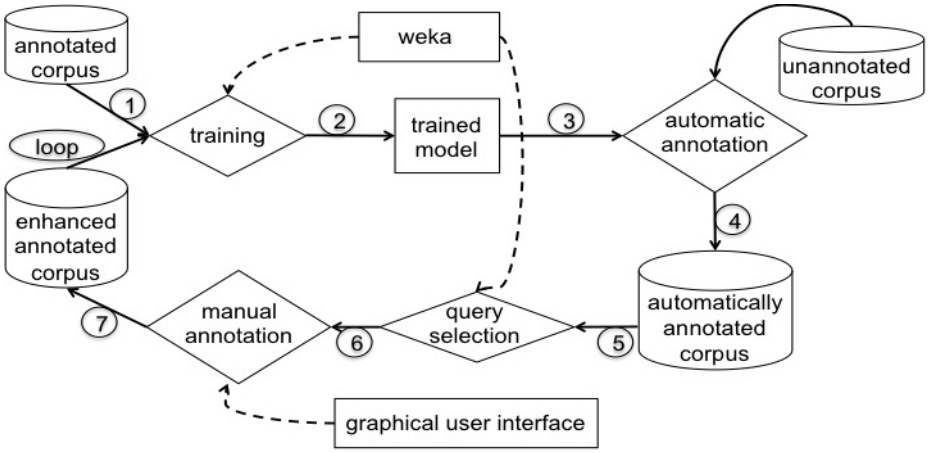

Fig. 1. Architecture of our basic License IE system, with the active learning component.

The system is prepared for multi-class annotation, because more than one property may be expressed in a single sentence.

5 Experimental setting

The goal of the experiments is to assess whether active learning strategies provide a faster learning curve than traditional methods, to alleviate portability of our approach to different domains. To that end, we compared active learning, with different parametrizations, with passive learning.

5.1 Dataset

We evaluated the performance of different learning methods using a manually annotated dataset of licenses. The corpus consists of the original labelled set of 37 licenses, and an unlabeled set of 396 licenses. It is composed of software licenses, source code licenses, data licenses, and content licenses; they are public as well as private domain licenses.

The labeled corpus has a total of 41,340 words, 2,660 of them unique. The mean of words per license is 1117.30, with a median of 400. This corpus has a total of 162 labelled instances, with a mean of 12.46 instances per class. The class with the most instances is *permission-to-distribute*, with a total of 33 instances, while

[13] `https://www.playframework.com/documentation/2.3.x/Home`

there are three classes with just one instance: *permission-to-read, prohibition-to-derive and requirement-to-attach-source*. Classes with very few instances are known to provide for very poor learned models, so we discarded classes with less than 5 labelled instances.

The training and evaluation corpus have been tagged previously and each instance was assigned to a single class. It must be noted that the majority of sentences in the corpus do not belong to any of the classes established by the ODRL vocabulary. In the classification setting, these examples belong to the class "null", which is actually composed of several heterogeneous classes with very different semantics, with the only common factor that their semantics are not captured by the ODRL vocabulary. The fact that this heterogeneous majority class is always present seems a good explanation for why the one-vs-all approach is more performant: it is easier to define one single class than some heterogeneous classes.

The unlabeled corpus is gathered manually, and has no overlap with the annotated corpus. This corpus has a total of 482,259 words, 8,134 unique. The mean of words per license is 1217.83, with a median of 505.50.

For the manual dataset annotation we adopted the CONLL IOB format., The B and I tags are suffixed with the chunk type according to our annotation task, e.g. B-PERMISSION, I-PERMISSION. We first tokenized the sentences using Stanford Parser [8], and we then added two columns, the first one for the annotation of the relation, and the second one for the value The Stanford Parser is also used to parse the instances of the unannotated corpus. From the unannotated corpus, sentences are taken as instances to be annotated by the automated classifier or the oracle.

5.2 Evaluation methods and metrics

The evaluation task is done with an automated simulation of the active learning loop on the annotated corpus. In this simulation, from the 156 original instances on the corpus, we started with an initial random set of 20 instances (roughly 12% of the annotated corpus). From this initial set the first model was learned, using the Multinomial Naïve Bayes approach. After that, the model was evaluated using 10-fold cross-validation.

With this initial model, we proceed to use the rest of the annotated instances as the unannotated corpus. With the data from the first model we carry out the selection of the queries from this "unannotated corpus" for manual annotation. In our experiments we try with three different approaches: queries of automatically annotated instances where the classifier is most certain sample, queries of instances where the classifier is most uncertain, and random selection (passive learning). The selected queries are then annotated using the provided information (as these queries are, in fact, from the annotated corpus) and added to the annotated corpus as new instances.

Once again the annotated corpus is used in a second iteration for creation and evaluation of a new model. The process is repeated until all the "unannotated"

instances are assigned their label. The number of newly annotated instances per iteration in our experiments is: 1, 3, 5 and 10.

The goal of this simulation is to show the steep of the curves in each one of the query selection methods in comparison to each other, with the highest slope being the best query selection strategy.

6 Analysis of results

In Figure 2 we can see the learning curves of our active learning approach, obtained as described in Section 5.2. We can see that the "most certain" strategy performs consistently better than the passive and most uncertain strategies, improving performance with fewer instances. The other two perform comparably if the number of instances added at each iteration is high, and the "most uncertain" approach performs even worse than the passive approach (random) if instances are added one at a time for each iteration. These results confirm our hypothesis that, for models inferred from very few training examples, maximizing the entropy of examples is not useful, while providing more evidence to define the core of the classes does provide an improvement in performance.

In an error analysis we can indeed see that the classes with most error are the smallest classes.This shows the benefit of growing the set of annotated examples, and thus the utility of an active learning approach for this task. The best strategy to grow from a very small dataset, with classes with very few instances, seems to be by choosing instances that are very similar to those already labelled, which provides a faster improvement in the performance of the classifier.

When examples are selected applying the "most uncertain" strategy are, they mostly belong to the "null" class, that is, they do not signal any of the classes relevant for the problem. Most of the sentences in licenses do not belong to any of the classes defined by the ODRL vocabulary and are classified as "null".

Providing examples for the class "null" is specially harmful for the resulting model for two main reasons. First, it grows the majority class, while small classes are kept with the same few examples, thus adding the problem of having an imbalanced dataset to the problem of having small classes with few instances. Second, the class "null" is composed by many heterogeneous classes that are not included in the ODRL vocabulary, and therefore its characterization is difficult and may be misleading.

Besides this configuration of classes, which can be found in very different domains, the domain of IE in licenses and normative text in general may be specially prone to an improvement of performance by labeling most certain examples first, because licenses and legal texts in general tend to be very formulaic, repeating the same wordings with very few variations, and small differences in form may signal differences in meaning, much more than in other domains, where differences in meaning are signalled by bigger differences in wordings.

Results for the best performances achieved by different passive learning approaches are summarized in Table 1. Those results were obtained using the whole dataset, corresponding to the rightmost extreme in the graphics of Figure 2.

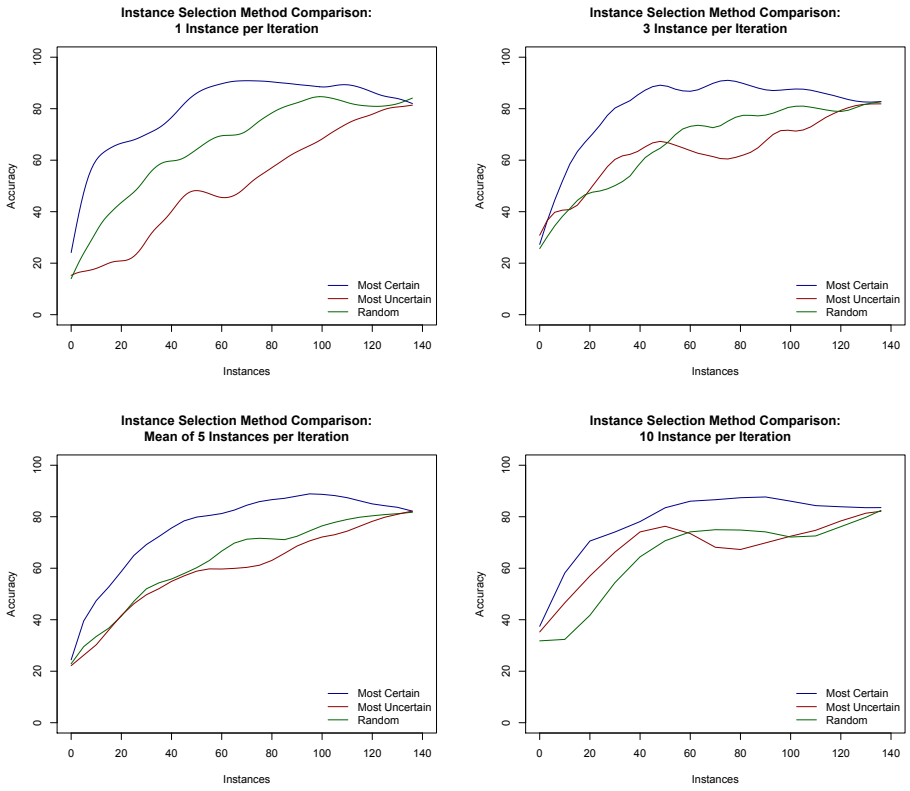

Fig. 2. Learning curves of active learning approaches with different policies for instance selection. In the y axis we depict accuracy, in the x axis, the number of instances added to training, and the different lines represent different strategies to pick the instances to be added in each iteration to the training corpus: random, ranked by most uncertain or by most uncertain.

7 Conclusions and Future Work

Dealing with legal information in the Web is a research area where several challenges need to be addressed. One of these challenges is the automated generation of machine-readable representations of the licenses, starting from their natural language formulation in legal documents. This issue is challenging not only because of the difficulties inherent to the task itself, but also due to the fact that the generated representation must be conformant with the starting regulation to avoid data misuse.

In order to overcome this problem, in this paper, we have developed a Web-based framework for active learning of instances and features, much in the spirit of Settles [12], but including features like multiclass labeling for a single instance

and using certainty of the classifier instead of Information Gain. Both the dataset and the new system are available online.

We have shown that, for the problem of inferring a classifier for normative text, where few labelled instances are available, active learning does provide a faster learning curve than traditional machine learning approaches, and it is thus an effective strategy to optimize human resources. It must be noted that in this specific setting, active learning is useful only if the most certain examples are selected to be hand-tagged, in contrast with the most frequent approach in active learning, called *uncertainty sampling*, where human annotators are given to annotate examples that the classifier is most uncertain about. This is caused by the fact that normative text is very formulaic thus tending to repetitions, but also to the fact that in this domain, slight differences in formulation tend to signal actual differences in meaning.

Several open issues have to be considered as future work. First, given the complexity of the task, our system provides an RDF representation of licenses considering their basic deontic components only, i.e., we model *permissions*, *prohibitions*, and *duties*. However, we plan to consider as future work further constraints expressed by the licenses, e.g., about time, payment information, and sub-licensing. Second, from the experimental setting perspective, we will explore some configurations that are left unexplored in this paper, like those classes with less than 5 labelled instances, with the most certain strategy to begin with. We will evaluate the performances of the system also using feature labeling by itself and in combination with instance labeling. We are currently exploring if there is a point in the development of the training set where it is more useful to switch from certainty sampling to uncertainty sampling, probably in correspondence with the different distributions of features in annotated and unannotated corpora. Finally, the system can be extended to a multilingual scenario (as far as the NLP pre-processing is available for the targeted languages), to provide machine readable versions of licenses published by national institutions, or licenses published in different languages.

References

1. Elena Cabrio, Alessio Palmero Aprosio, and Serena Villata. These are your rights - A natural language processing approach to automated RDF licenses generation. In The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings, pages 255–269, 2014.
2. Chih-Chung Chang and Chih-Jen Lin. Libsvm - a library for support vector machines, 2001. The Weka classifier works with version 2.82 of LIBSVM.
3. Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI'05, pages 746–751. AAAI Press, 2005.
4. Dmitriy Dligach and Martha Palmer. Good seed makes a good crop: Accelerating active learning using language modeling. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11, pages 6–10, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

5. Pinar Donmez, Jaime G. Carbonell, and Paul N. Bennett. Dual strategy active learning. In Joost N. Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, ECML, volume 4701 of Lecture Notes in Computer Science, pages 116–127. Springer, 2007.
6. G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 81–90. ACL, 2009.
7. Michael Kearns. Efficient noise-tolerant learning from statistical queries. J. ACM, 45(6):983–1006, November 1998.
8. Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
9. David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In In Proceedings of the Eleventh International Conference on Machine Learning, pages 148–156. Morgan Kaufmann, 1994.
10. Jay Pujara, Ben London, and Lise Getoor. Reducing label cost by combining feature labels and crowdsourcing. In ICML Workshop on Combining Learning Strategies to Reduce Label Cost, 2011.
11. B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
12. B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1467–1478. ACL, 2011.
13. B. Settles. Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.
14. B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1069–1078. ACL, 2008.
15. Christopher T Symons and Itamar Arel. Multi-View Budgeted Learning under Label and Feature Constraints Using Label-Guided Graph-Based Regularization. 2011.
16. Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.
17. Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pages 1137–1144. Association for Computational Linguistics, 2008.