Turning Silver into Gold: Error-Focused Corpus Reannotation with Active Learning

Pierre André Ménard, Antoine Mougeot
Centre de recherche informatique de Montréal
pierre-andre.menard@crim.ca, antoine.mougeot@crim.ca

Abstract

While high quality gold standard annotated corpora are crucial for most tasks in natural language processing, many annotated corpora published in recent years, created by annotators or tools, contains noisy annotations. These corpora can be viewed as more *silver* than *gold* standards, even if they are used in evaluation campaigns or to compare systems' performances. As upgrading a silver corpus to gold level is still a challenge, we explore the application of active learning techniques to detect errors using four datasets designed for document classification and part-of-speech tagging.

Our results show that the proposed method for the seeding step improves the chance of finding incorrect annotations by a factor of 2.73 when compared to random selection, a 14.71% increase from the baseline methods. Our query method provides an increase in the error detection precision on average by a factor of 1.78 against random selection, an increase of 61.82% compared to other query approaches.

1 Introduction

As machine learning is increasingly predominant in natural language processing tasks, the need for annotated data, and more specifically linguistic annotations, intensify greatly. Tasks like bias detection, named entity detection (NER) and recognition, part-of-speech (POS) tagging, semantic role labeling, assessment evaluation in discourse, dependency parsing and sentiment analysis can all be modeled as machine learning problems. They require a large amount of human annotated information to enrich raw textual resources. Creating large annotated resources for these tasks requires a lot of time and effort to insure a carefully planned and well-defined annotation protocol, obtain and encode expert knowledge, do curation steps and so on. Even when using highly qualified human annotators, errors can always make their way into the final annotated corpus. Another way to obtain annotations is to apply existing state-of-the-art algorithms (also trained on previously annotated data) on a raw corpus.

While some researchers publish new versions of their hard earned annotated resources based on the feedback obtained from users, others often have to set them aside to work on other projects and resources, leaving the corpora with their original errors. One way to improve these resources without the same level of effort would be to reannotate them using an active learning process to quickly find errors and discrepancies and resubmit them to expert annotators.

This article relates an experiment which explores the potential application of active learning for corpus reannotation, which context is detailed in Section 2 with related works listed in Section 3. Two new approaches and multiple baselines are then explained in Section 4, followed by the list of datasets used (Section 5) and the experiments combining all these elements (Section 6). We then comment on the experiment (Section 7) and close with a review of the work done and some future works in the final section.

2 Context

This section gives an overview of the two main aspects of this research, namely corpus reannotation and active learning, with their challenges and possibilities. The current work is at the crossroad of these two topics.

Proceedings of Recent Advances in Natural Language Processing, pages 758–767,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_088

2.1 Corpus Reannotation

The errors present in annotated corpora can have many sources. They can be simple typographical errors created during transcription or from the begining on uncurated information. On a higher cognitive level, another frequent cause of error is annotators' discrepancies over expert knowledge, in which disagreement over labels are not solved in a consistent way. Protocol inconsistencies are yet another source of noise in annotated corpora, often encountered when unforeseen cases fail to be brought to light, managed, or added to the protocol. Moreover, there is also all the possible causes from the annotator side, like misunderstanding the protocol, errors caused by fatigue, solving ambiguities with the most favorable case instead of reporting it, and many others. All these issues can diminish the quality of the annotations.

The nature of an annotation task can also influence the risk of generating errors. Intuitively, fine-grained classifications are often considered more error-prone than those with a small number of class values. This can be attributed to fuzzy frontiers between close-by values, overlapping values (i.e. choosing between "entertainer" and "comedian") or failing to recall a specific classification option in a very large set of values.

Errors might also occur based on the interpretation of different annotators on the same case. This can be the case in named entity classification tasks, when an occurrence can be considered both an organization and a location (i.e. "I went to register at the office of University X"). These are edge cases that are often overlooked in annotation protocols and might result in inconsistencies in the final annotated corpus.

A silver standard corpus (Rebholz-Schuhmann et al., 2010) is usually defined as a noisy set of ground truth annotations provided automatically by state-of-the-art algorithms, while *gold* labels are the higher quality annotations created by expert annotators. The silver labels are normally produced manually by human agents, but can also be obtained automatically by tools or trained prediction models. Of course, the gold labels might still contain some degree of noise, but they are considered of better overall quality than the silver version.

Corpus reannotation can be a tedious undertaking, as it not only requires the same expert domain knowledge to correct the annotations, but also a deep understanding of the original protocol, often created by another team. It may also require to modify the protocol or classification values, and also require an additional effort of the annotators to assess if they are not creating more errors when modifying an original answer. For all these reasons, it is important to explore ways by which some of the effort might be lowered, such as the application of active learning on noisy corpora.

2.2 Active Learning

Active learning (Cohn et al., 1996) has been used to lower the annotation effort needed to train a prediction model in natural language processing tasks. It traditionally involves a dialogue between one (or more) human annotator(s) and a machine learning algorithm, the former being proficient at annotating instances that the latter is providing based on relevance measures. The process can be split into three distinct steps.

The first one, called *seeding*, is where the algorithm must choose instances without being able to rely on any annotation from the expert. The chosen instances are submitted to be annotated by the expert. This starts the second and longest step, the *querying* phase. The active learning engine iterates between training a prediction model with the gold values (training set) from the expert, choosing new instances relevant to training a better model, submitting them to the expert and adding the expert's answers back into the training set. The third step, *stopping*, is applied after each training of the querying phase. It checks if there is enough information in the model to annotate the rest of the corpus automatically so that the annotation can stop. A bad stopping criterion might overfit the model, lowering its predictive power.

One challenge of active learning is to balance between specializing the classification on known cases (for example, annotation errors), thus improving the performance for current classes, or exploring the problem space to find unknown but relevant instances that could improve the overall performance of the model.

The goal of the current experiment is not to produce the best prediction model, but to explore if the reannotation effort on a corpus can be reduced by using the active learning process with different algorithms. These algorithms should not target the most informative instances for a prediction model, but instead choose those that are more likely to be

errors. If successful, the model should then be able to extract more errors as it progresses.

3 Related Works

While different flavors of reannotation have been used for natural language processing tasks on text corpora, the work of (Rehbein and Ruppenhofer, 2017) is the most similar to our contribution from an error finding perspective with active learning. They evaluate the query-by-committee (QBC) and variational inference (VI) active learning methods to perform error detection on part-of-speech tagging and named entity recognition (NER) tasks. These approaches were tested in four contexts: in-domain (same type of training and testing data) on English POS, out-domain (different training and testing) on English POS, new task and language on German NER, and real-world context with human annotators on POS task. After 1,000 iterations, the VI approach generally gives a better noise reduction than QBC. While the POS task is similar to the one used in our experiment (although on a different language), the NER task uses only 4 class values (location, organization, person and miscellaneous) while documents in our classification task can be categorized into a high number of types. They do not specify the experimental seeding methods used to choose the first examples.

Another similar research is Skeppstedt (2013) which uses active learning with two sources of tools generated annotation in order to tag and classify named entities in Swedish clinical text documents. The challenge of combining multiple pre-annotated sources and active learning is to provide the right quality of information. If the sources are too noisy, the task will be more difficult and unreliable. On the other hand, providing high-quality sources might lower the attention and interest of the annotators. The proposed method tries to overcome these two points by showing the sources without specifying which one is most likely to be correct. No performance evaluation was done for the proposed approach.

Other experiments make usage of preannotated information without applying active learning methods, like (Chou et al., 2006) who use a semantic role labeling tool trained on PropBank to pretag a biomedical corpus called BioProp. After the automatic annotation step, a human annotator manually checks the silver values and corrects them as needed. Other reannotation efforts may be conducted by adding human resources to the task to distribute the effort among multiple users. This is the proposition made in (Hovy et al., 2014) by applying crowd sourced reannotation. In the same multiuser settings, (Lin et al., 2016) propose an approach to reannotate labels by integrating another human oracle in order to improve the quality of the annotations.

4 Methods

In order to improve noise detection, we focus on the seeding and querying steps of the active learning process. As the goal is to facilitate annotation of incorrect annotation, and not to create a prediction model per se, we did not explore the stopping phase. The following sections detail the baselines as well as the new methods used for the experiments in Section 6.

4.1 Seeding Methods

The seeding method's main goal from a reannotation perspective is to provide the highest error ratio for the budgeted seed size. This contrasts with a usual active learning task which is to find the most informational instances to annotate in order to improve the model's performances.

To capitalize on the silver classification information, we used outlier detection methods separately on each unique silver value. Our hypothesis for this is that most of the annotations should be of good quality, although noisy, meaning that clustering the instances for a single silver value should produce one or many clusters of correctly classified instances. A large enough dataset should then provide a cluster for each valid manifestation (depending on the features used) of a silver value. As the clusters represent valid cases, the outliers should represent either rare cases or, ideally, noisy labels which should be reannotated by the expert.

For each silver value of a corpus, a random instance was chosen in the detected outliers, to test the hypothesis that they should mostly be incorrectly annotated cases.

Four other outlier detection methods were tested as baselines to assess their performances and compare them to the above method. These are not normally used in the seeding phase and they do not consider the presence of a silver value in the feature set. The first is the one-class SVM (Schölkopf et al., 2001) which trains a support vector machine with a radial basis kernel function

and returns the lowest supported instances in a distribution.

The local outlier factor (Breunig et al., 2000) computes the local deviation of density for an instance with its closest neighbors using a k-nearest approach. If the local density of the close-by instances is significantly higher, the instance is considered an outlier and is selected for human annotation.

The isolation forest algorithm (Liu et al., 2008) detects outliers by selecting the most isolated instance of a randomly selected feature set and split point in the value range. These splits are then projected into a tree structure and the average path length to an instance gives its degree of isolation, choosing the highest degree of isolation as the best potential instance to annotate.

Finally, the covariance detector (Rousseeuw and Driessen, 1999) uses a Gaussian distribution around the density cluster to assess the degree to which an instance might be part of that cluster.

4.2 Querying Methods

The proposed *double centroid* approach is based on the hypothesis that an annotator, either human or tool, tend to produce similar types of errors through the annotation process. The source of these types could be the inability to differentiate between two classes, unknown terminology, etc.

The method first use density-based clustering to group together newly annotated instances from the seeding or previous querying steps. It is applied once on erroneous instances, where the silver and new gold values did not match, and once on non-erroneous instances, where the silver and new gold values matched. This gives multiple clusters containing either noisy (C_n) or matching (C_m) annotations.

$$rank(l) = \left| \sum_{i=1}^{C_n} \frac{|C_i|}{dist(l, C_i)^2} - \sum_{j=1}^{C_m} \frac{|C_j|}{dist(l, C_j)^2} \right| \quad (1)$$

As shown in equation 1, for each silver instance l that was not yet picked for relabeling, a weighted squared distance $dist(l, C)^2$ is calculated separately against each cluster centroid. The weight used is the cluster cardinality $|C|$, so that nearer and larger clusters have more influence on an instance. These weighted distances are then summed up with opposing values, in this case C_n clusters having a positive influence and C_m having a negative one. The algorithm then selects those that have the highest value, following the hypothesis that they would be similar to known errors.

Other methods often applied in a standard active learning process have also been used as a basis of comparison with the proposed method, namely *distance to centroid*, *cosine similarity*, *hierarchical clustering* and *margin query*.

Distance to centroid calculates a density center point (centroid) from each instance of a specific gold value asked from the expert annotator. Each instance is then checked against each centroid and the ones which are furthest from all points are selected.

The cosine similarity works in a similar way as the previous method but uses the cosine between the instance and the centroids to assess the proximity.

Hierarchical clustering also uses the distance to clusters as a degree of uncertainty to choose ambiguous instances, but goes one step further by splitting these instances to choose only those which are the most different from one another. This usually helps to provide a better sample to annotate and avoid ambiguous but very similar instances.

Margin query chooses the instances with the smallest difference between the most probable predicted class and the second most probable.

5 Datasets

For this experiment, we used a total of four datasets, two targeting a document classification task and two for a text sequence classification task on part-of-speech tags.

As there were no publicly available manually corrected datasets, and correcting an existing one would have been too time consuming for the scope of this project, we simulated the noise level by applying a classification tool to each manually annotated corpus. They are each presented in the following sections and Table 1 shows the size and error rate for each of them. The error rate is calculated by dividing the number of errors in the silver standard (compared to the gold standard) by the total number of elements. Sections 5.5 and 5.6 describe how the noise was generated for each corpus to calculate the error rate.

Corpus	Size	Error rate
Reuters	9,149 docs	0.0941
WoS	46,985 docs	0.3916
GSD-French	402,120 words	0.1015
Sequoia	70,572 words	0.1080

Table 1: Corpus size (documents or words) and error rate.

5.1 WOS-46895

The WOS corpus (Kowsari, Kamran et al., 2018) contains 46,985 documents in English taken from the Web of Science website. Each document contains the abstract from a published scientific paper. These documents were picked from the fields of psychology, medical sciences, biochemistry, computer science and three specialization of engineering. While each document is only classified in a single topic, terminology coverage of some topics overlaps with others, such as between biochemistry and medical sciences.

Each topic is further broken down into 134 specialized areas, varying from 9 to 53 areas for each topic. While this dataset was primarily created for hierarchical classification, we only used the 134 areas (the second layer of classification) for the purpose of this research.

5.2 Reuters

The Reuters-21578 corpus (Lewis, 2004) is composed of 10,788 English documents consolidated for the text classification challenge of document understanding conference (DUC). There are 90 categories in the corpus. However, that dataset was originally used for multi-class classification. To use it for our research, we extracted only the articles having a single class and kept only the classes that appeared more than once in the dataset. The final dataset is made of 9,149 instances across 56 categories like *acq* and *earn*. The labels' distribution is heavily skewed as two of these classes make up 75% of the dataset instances.

5.3 French-GSD

We used the French portion of the GSD corpus (McDonald et al., 2013), version 2.2 at the time of writing. We merged the three parts (dev, train, test) into a single corpus consisting of 402,426 words (16,448 sentences). While it is fully tagged with dependence trees, we only retained the 17 universal dependency-based part-of-speech tags from the open class (*ADJ*, *ADV*, *INTJ*, etc.), the closed class (*ADP*, *AUX*, *CCONJ*, *DET*, etc.) and the other class (*PUNCT*, *SYM*, *X*). The feature set for each instance included the token's position in the sentence, surface form, lemma, length, presence of space after the token and case type (capitalized, all capital letters or mixed case).

5.4 Sequoia

The Sequoia corpus (Candito and Seddah, 2012) contains 3,099 French sentences (70,572 tokens) taken from different corpora such as Europarl, *L'Est Republicain* newspaper, French Wikipedia and European Medicine Agency. It has initially been annotated with constituency trees and then converted to surface syntactic dependency trees. In our experiment, we only classified the part-of-speech (POS) information. The feature set was the same as the French-GSD corpus.

5.5 POS Processing and Vectorisation

The GSD and Sequoia corpora were initially tagged manually with universal part-of-speech tags[1] which we used as the gold standard. While dependency information was widely available in the original corpus, we removed them as they would not be available in a raw text corpus without applying a high quality dependency analyzer. We kept morphological features as listed in Section 5.3.

In order to create a silver version of the dataset, each token was automatically reannotated with TreeTagger (Schmid, 1997) to provide a new set of tags. These tags were then converted to the universal part-of-speech tagset to make them comparable with the original corpus tags.

These corpora were then vectorized to be processable by the classification algorithms, projecting the information of each instance in a feature space. The feature set was also enriched for each instance with the information from the last five tokens and the next five tokens. The sentence boundaries were respected, so that the first token in a sentence would only get the next five tokens, the second token would get the previous token and the next five, and so on. As the algorithms used cannot deal with nominal data, each feature was one-hot encoded. Once encoded, each dataset contained 1,156 features including the silver annotation.

[1] https://universaldependencies.org/u/pos/

5.6 Classification Preprocessing and Vectorisation

The text documents in the WOS and Reuters corpora were stemmed using Porter stemmer (Porter, 1997) and had their stop words removed. In order to simulate silver level annotations, the corpora were vectorized and annotated using a five-parts iteration. In other words, the corpus was split into five batches containing 20% of the corpus, a model was trained on the gold values of 80% of the corpus, recreating a new model to annotate the remaining 20%, and a different batch was swapped to be annotated each time.

The two corpora were then vectorized with two word embeddings methods, fastText (Bojanowski et al., 2016), Word2vec (Mikolov et al., 2013), and the tf-idf (Salton and McGill, 1986) statistical method. The feature set size of the output vector from each method, either produced with neural networks or statistical measure, was set to 1,000 features.

Corpus	Method	F1
Reuters	fastText	0.87
	Word2vec	0.88
	tf-idf$_3$	0.91
WOS-46895	fasttext	0.59
	Word2vec	0.57
	tf-idf$_3$	0.61

Table 2: Performances of vectorization methods.

To see which one provided the best expressivity with its features, the vectors for each corpus were evaluated with ten-fold validation using logistic regression as the passive learning algorithm. The results in Table 2 show the performances for each vectorization method on each corpus. While the results are quite close to one another, the statistical tf-idf method using unigrams to trigrams (Tf-idf$_3$) provides higher results on all datasets. For this reason, we use the feature set provided by this method for the experiments.

6 Experiment

6.1 Seeding Experiment

We use the error rate of the corpus (as previously reported in Table 1) as a random baseline for seeding, which is the equivalent of making a random selection. As studied in (Hu et al., 2010), most of the papers either use this type of seeding method for active learning, choose a fixed number of instances from each target class value (knowledge that would not be accessible in a real world setting) or simply fail to mention the method.

The methods described in the previous section were run 100 times to smooth out the randomness effect of some outlier selection methods. It also helped to show if most of the outliers were in fact valid errors to be detected. Each chosen outlier was then compared to the gold value to verify if it was an error or a valid annotation. The number of unique silver values were not scaled down to a specific seed size so as to provide an overall measure of performance.

As some clustering methods do not scale well with large datasets, we applied a feature reduction algorithm on each dataset. We used a random forest (Breiman, 2001) estimator using 100 trees, building each one with a random subset of features from the dataset, evaluating the relevance of each feature and discarding the less meaningful ones. The Sequoia and French-GSD corpora were respectively reduced to 261 and 223 features, while 250 and 334 features were retained from the Reuters and WOS-46895 corpora. The silver annotation was used as the relevance indicator for this process instead of the gold which would not be available in a real setting.

The results presented in Table 3 show an improvement across all methods compared to baseline performances. The error rate column is averaged from all the results in the experimental runs. The gain is the ratio between the average error rate and the baseline performance from Table 1 (error rate column) using a random selection or a standard seeding method.

While all seeding methods provide an improvement compared to the baseline, the local outlier factor method seems the most promising when assessed from the average gain global score over all corpora. Performances on the WOS corpus were not favored by any of the four methods and were just marginally better with the local outlier factor.

6.2 Querying Experiment

The four baseline methods detailed in Section 4 were applied to each vectorized corpus for the document classification and POS tagging tasks. To avoid boosting the performance of the error seeking query method, we did not use the previous approaches for seeding. Instead we randomly se-

Method	Measure	Reuters	WoS	Sequoia	French-GSD	Average gain
SVM	EDP	0.3576	0.4346	0.1429	0.2938	
	Gain	3.80	1.11	1.32	2.89	2.28
Covariance	EDP	0.3606	0.4331	0.1786	0.2971	
	Gain	3.83	1.11	1.65	2.93	2.38
Isolation forest	EDP	0.3424	0.4336	0.1929	0.2267	
	Gain	3.64	1.11	1.79	2.23	2.19
Local outlier	EDP	0.4030	0.4369	0.2000	0.3729	
	Gain	4.28	1.12	1.85	3.67	**2.73**

Table 3: Seeding phase error detection precision and gain.

lected a subset of 20 instances for the seeding step. This ensured that the initial training set would have about the same standard level of errors as in the rest of the corpus. The query phase selected 20 instances per iteration before retraining the random forest model. These queries were answered by an artificial annotator who had access to the corresponding gold values. Each method was run on the corpora 20 times to smooth out the variability of the random aspects of some methods.

Table 4 shows the average error detection precision (EDP) for the complete set of experiments. It is the ratio between the number of errors detected (where silver and gold labels does not match) and the total number of queried instances at a specific point. A score of 1 would mean that the algorithm only submitted errors to be reannotated by the oracle. As the active learning process can be stopped at any point, the recall score is not used as it would imply that all errors should be submitted before the end of the process.

The gain ratio is the error detection precision divided by the corresponding corpus error rate. Gain values lower than one mean a lower performance than random selection. The EDP shown was calculated after 200 annotations, meaning one seed and 19 query iterations. This number was choosen from real-world experiences, where annotators usually consider that some errors should be encountered at that point, if any are to be found.

We can see that the proposed double centroid method outperformed all the other approaches on every corpus. Most methods did not perform well on the WoS corpus, which incidentally had four times the error rate of other corpus. The best gain ratio of our method was when detecting POS tagging errors on the FR-GSD and classification

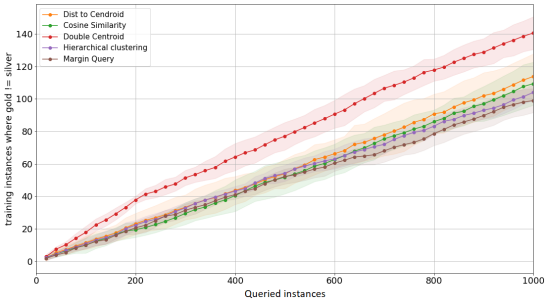

Figure 1: Performances of querying methods on the Reuters corpus.

errors on the Reuters corpora. Aside from double centroid, most other methods performed at the same level as the others, except for distance to centroid on the Reuters corpus.

As seen on Figure 1, which shows all methods applied to the classification task on the Reuters corpus for a total of 1,000 instances annotated, all methods seem to have a relatively stable slope. The double centroid method loses some velocity after hitting 220 instances. The other methods diverge near the end, but not very distinctively.

7 Analysis and Discussion

Looking at the slopes in Figure 1 at the start of the process, we can see that they start at about the same level of performance before settling into their tendencies. This is mainly due to the fact that the seed does not contain many instances about different gold values to provide significant clusters.

Why the performance drops on the WoS corpus compared to Reuters or other corpora can be attributed in part to the low expression power of the vectorized feature set shown in Table 2. They pro-

Method	Measure	Reuters	WoS	Sequoia	French-GSD	Average gain
Distance to centroid	EDP	0.1194	0.3954	0.1049	0.1196	
	Gain	1.27	1.01	1.03	1.11	1.10
Cosine similarity	EDP	0.0974	0.3796	0.1198	0.1003	
	Gain	1.04	0.97	1.18	0.93	1.03
Hierarchical clustering	EDP	0.1110	0.3943	0.1196	0.1079	
	Gain	1.18	1.01	1.18	1.00	1.09
Margin query	EDP	0.1032	0.3937	0.1074	0.1057	
	Gain	1.10	1.01	1.06	0.98	1.03
Double centroid	EDP	0.1982	0.4283	0.2499	0.1565	
	Gain	2.11	1.09	2.46	1.45	**1.78**

Table 4: Query phase error detection precision and gain after 200 instances.

vided only two thirds of the performance on the passive learning task compared to those produced from the Reuters corpus. While pretrained word embeddings could have been used, we wanted to avoid the issue of unknown tokens when dealing with specialized corpora.

In order to lower even more the effort of annotators, the next logical step would be to validate if the final prediction model was either as good or better at predicting any type of instances on the remaining corpus when compared to a standard, non-error detecting active learning process. This would entail that an error detection model could simply be used to annotate the remaining instances, correcting more errors and minimizing the creation of additional noise.

8 Conclusion and Future Work

We experimented with active learning methods on noisy corpora to lower the noise level, thus improving the overall quality of the datasets. Our proposed seeding method targeting error detection provided a gain factor of 2.73 when compared to the most used random seeding in active learning, while our double centroid method provided a 61.82% increase for finding noisy annotations when compared to baseline approaches used for active learning.

While there is room for improvement, these results show the potential application of active learning for corpus annotation. The applicability on two NLP tasks on different units (words or documents) shows a good adaptability of the tested methods, as the use of two languages in the corpora to avoid relying on language-specific techniques.

One untested hypothesis is if these methods provide the same level of performance when applied to corpora with human-generated noise instead of tool-generated noise. We expect that noise level might be lower in published and broadly used datasets if reannotated with the original protocol. This might influence the effectiveness of the tested methods.

These approaches should be tested on a broader set of natural language processing tasks, such as sentiment analysis, information quality, relevance identification, named entity classification, etc. This would either help to further advance the demonstration about the effectiveness of these methods, or to develop new approaches to facilitate the task of reannotation.

Some influential aspects of active learning have been left aside in this experiment as they did not directly implicate human annotators, like the cognitive charge of annotation correction for a human agent. This requires not only to assess the context of the existing annotation to deduce the correct annotation, but also to ponder, when the existing annotations differ, if they are not adding more noise.

Acknowledgments

This work has been supported by the Ministère de l'Économie et de l'Innovation du Québec (MEI). We would like to thank the anonymous reviewers for their relevant comments.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Leo Breiman. 2001. Random forests. In *Machine Learning*. pages 5–32.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *ACM SIGMOD 2000 Int. Conf. On Management of Data*. page 12.

Marie Candito and Djamé Seddah. 2012. Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*. ATALA/AFCP, pages 321–334. http://aclweb.org/anthology/F12-2024.

Wen-Chi Chou, Richard Tzong-Han Tsai, Ying-Shan Su, Wei Ku, Ting-Yi Sung, and Wen-Lian Hsu. 2006. A semi-automatic method for annotating a biomedical proposition bank. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*. Association for Computational Linguistics, Sydney, Australia, pages 5–12. https://www.aclweb.org/anthology/W06-0602.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *J. Artif. Int. Res.* 4(1):129–145.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*. pages 377–382. http://aclweb.org/anthology/P/P14/P14-2062.pdf.

Rong Hu, Brian Mac Namee, and Sarah Jane Delany. 2010. Off to a good start: Using clustering to select the initial training set in active learning. In *FLAIRS Conference*.

Kowsari, Kamran, Brown, Donald, Heidarysafa, Mojtaba, Jafari Meimandi, Kiana, Gerber, Matthew, and Barnes, Laura. 2018. Web of Science Dataset. https://doi.org/10.17632/9rw3vkcfy4.6.

David Lewis. 2004. Reuters 21578 data set version 1.0 http://www.daviddlewis.com/resources/ testcollections/reuters21578/readme.txt.

Christopher H Lin, M Mausam, and Daniel S. Weld. 2016. Re-Active Learning: Active Learning with Relabeling. In *Thirtieth AAAI Conference on Artificial Intelligence*. page 8.

F. T. Liu, K. M. Ting, and Z. Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. pages 413–422. https://doi.org/10.1109/ICDM.2008.17.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 92–97. http://aclweb.org/anthology/P13-2017.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

M. F. Porter. 1997. Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, chapter An Algorithm for Suffix Stripping, pages 313–316. http://dl.acm.org/citation.cfm?id=275537.275705.

Dietrich Rebholz-Schuhmann, Antonio José Jimeno-Yepes, Erik M. van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Katrin Tomanek, Elena Beisswanger, and Udo Hahn. 2010. The CALBC silver standard corpus for biomedical named entities — a study in harmonizing the contributions from four independent named entity taggers. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Languages Resources Association (ELRA), Valletta, Malta. http://www.lrec-conf.org/proceedings/lrec2010/pdf/888_Paper.pdf.

Ines Rehbein and Josef Ruppenhofer. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1160–1170. https://doi.org/10.18653/v1/P17-1107.

Peter J. Rousseeuw and Katrien Van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41(3):212–223. https://doi.org/10.1080/00401706.1999.10485670.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval .

Helmut Schmid. 1997. Probabilistic part-of-speech tagging using decision trees. In Daniel Jones and

Harold Somers, editors, *New Methods in Language Processing*, UCL Press, London, GB, Studies in Computational Linguistics, pages 154–164.

Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Comput.* 13(7):1443–1471. https://doi.org/10.1162/089976601750264965.

Maria Skeppstedt. 2013. Annotating named entities in clinical text by combining pre-annotation and active learning. In *Proceedings of the ACL Student Research Workshop*. page 74–80.