Identifying Key Sentences for Precision Oncology Using Semi-Supervised Learning

Jurica Ševa, Martin Wackerbauer and Ulf leser
Knowledge Managment in Bioinformatics
Humboldt Universität zu Berlin
Berlin, Germany
{seva,wackerbm,leser}@informatik.hu-berlin.de

Abstract

We present a machine learning pipeline that identifies key sentences in abstracts of oncological articles to aid evidence-based medicine. This problem is characterized by the lack of gold standard datasets, data imbalance and thematic differences between available silver standard corpora. Additionally, available training and target data differs with regard to their domain (professional summaries vs. sentences in abstracts). This makes supervised machine learning inapplicable. We propose the use of two semi-supervised machine learning approaches: To mitigate difficulties arising from heterogeneous data sources, overcome data imbalance and create reliable training data we propose using transductive learning from positive and unlabelled data (PU Learning). For obtaining a realistic classification model, we propose the use of abstracts summarised in relevant sentences as unlabelled examples through Self-Training. The best model achieves 84% accuracy and 0.84 F1 score on our dataset.

1 Introduction

The ever-growing amount of biomedical literature accessible online is a valuable source of information for clinical decisions. The PubMed database (National Library of Medicine, 1946-2018), for instance, lists approximately 30 million articles' abstracts. As a consequence, machine learning (ML) based text mining (TM) is increasingly employed to support evidence-based medicine by finding, condensing and analysing relevant information (Kim et al., 2011). Practitioners in this field search for clinically relevant articles and findings, and are typically not interested in the bulk of search results which are devoted to basic research. However, defining clinical relevance in a given abstract is not a trivial task. On top, although abstracts provide a very brief summary of their corresponding articles' content, practitioners determine abstracts' clinical relevance based on only a few key sentences (McKnight and Srinivasan, 2003). To optimally support such users, it is thus necessary to first retrieve only clinically relevant articles and next to identify the sentences in those articles which express their clinical relevance.

Any survival benefit of dMMR was lost in N2 tumors. Mutations in BRAF(V600E) (HR, 1.37; 95% CI, 1.08 to 1.70; P = .009) or KRAS (HR, 1.44; 95% CI, 1.21 to 1.70; P ¡ .001) were independently associated with worse DFS. The observed MMR by tumor site interaction was validated in an independent cohort of stage III colon cancers (P(interaction) = .037).

Example 1: Snippet of highlighted clinically relevant (or key; yellow background color) and irrelevant (no background color) sentences in a precision oncology setting. Source document with PMID 24019539.

In this work, we present an ML pipeline to identify key (clinically relevant) sentences, in a precision oncology setting, in abstracts of oncological articles to aid evidence-based medicine. This setting is implied throughout the text when referring to clinical relevance or key (clinically relevant) sentences. An example of relevant and irrelevant sentences is shown in Example 1. For solving this problem no gold standard corpora is available. Additionally, clinical relevance has

Proceedings of the BioNLP 2018 workshop, pages 35–46
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

only a vague definition and is a subjective measure. As manually labelling text is expensive, semi-supervised learning offers the possibility to utilize related annotated corpora. We focus on Self-Training (Wang et al., 2008), which mostly relies on supervised classifiers trained on labelled data and use of unlabelled examples to improve the decision boundary. Several corpora can be used to mitigate the issues arising from the lack of gold standard data set and data imbalance. These corpora implicitly define characteristics of key sentences, but cannot be considered as gold standards. In the following, we call them "silver standard" corpora - collections of sentences close to the intended semantic but with large amounts of noise. Specifically, we employ *Clinical Interpretations of Variants in Cancer* (CIViC) (Griffith et al., 2017) for implicit notion of clinical relevance and positive data points, i.e. sentences or abstracts which have clinical relevance. Unfortunately, negative data points, i.e. sentences or abstracts which do not have clinical relevance, are not present in this data set. *PubMed abstracts*, referenced by CIViC, are used as unlabelled data. Since we consider all sentences in CIViC to be positive examples and the corresponding abstracts are initially unlabelled, additional data for negative examples is required. We utilize the *Hallmarks of Cancer Corpus* (HoC) (Baker et al., 2016) as an auxiliary source of noisy labelled data. To expand on our set of labelled data points we propose transductive learning from positive and unlabelled data (PU Learning) to identify noise within HoC, with CIViC as a guide set for determining the relevance of sentences from HoC. This gives us additional, both positive and negative data points, used as an initialization for Self-Training. The pipeline is available at https://github.com/nachne/semisuper.

2 Related Work

Sentence classification is a special case of text categorisation. It has been used in a wide range of fields, like sentiment analysis (Yu and Hatzivassiloglou, 2003; Go et al., 2009; Vosoughi et al., 2015), rhetorical annotation, and automated summarisation (Kupiec et al., 1995; Teufel and Moens, 2002). Between the two, feature engineering has been reported as the major difference. For instance, common stop words like "but", "was", and "has" are often among the top features for sentence classification, and verb tense is useful to determine a sentence's precise meaning (Agarwal and Yu, 2009; Khoo et al., 2006). Additional features beyond the pure language level have also been proposed. For sentiment analysis, Yu and Hatzivassiloglou (2003) use a dictionary of semantically meaningful seed words to estimate the likely positive or negative polarity of co-occurring words, from which in turn a sentences' polarity is determined. Teufel and Moens (2002) focus on identifying rhetorical roles of sentences for automatic summarisation of scientific articles. They use sentence length and location, the presence of citations and of words included in headlines, labels of preceding sentences, and predefined cue words and formulaic expressions accompanying Bag of Words (BOW).

Text represented as high dimensional BOW vectors has been reported to be often linearly separable, making Support Vector Machines (Joachims, 1998) (SVM) a popular choice for classifiers. Conditional Random Fields (CRF) have been used to predict sequences of labels rather than labelling sentences one by one (Kim et al., 2011). In recent years, Neural Networks (NN) and Deep Learning (DL) has increasingly been used, e.g. using Convolutional Neural Networks (CNN) (Kim, 2014; Rios and Kavuluru, 2015; Zhang et al., 2016; Conneau et al., 2017). Other authors employ various versions of Recurrent Neural Networks (RNN): LSTM (Hassan and Mahmood, 2017), bi-directional LSTM (Dernoncourt et al., 2017; Zhou et al., 2016) or convolutional LSTM (Zhou et al., 2015). The use of DL has also popularised the use of pre-trained word embedding vectors. Habibi et al. (2017) show that the use of word embeddings, in general, increases the quality of biomedical named entity recognition pipelines.

Specific to the biomedical domain, sentence classification has been used to determine the rhetorical role sentences play in an article or abstract. Ruch et al. (2007) propose using the "Conclusion" section of abstracts as examples for key sentences. McKnight and Srinivasan (2003) have classified sentences in abstracts of randomised control trials as belonging to the categories "Introduction", "Methods", "Results", and "Discussion" (IMRaD), using section headlines as soft labels for training data in addition to a smaller hand annotated corpus. They also report that adding sentence location as a feature improved performance on the "Introduction" and "Discussion" categories. Kim et al. (2011) used a CRF for sequential classification, trained on the hand-annotated Population, Intervention, Background, Outcome, Study Design of evidence-based medicine, or Other (PIBOSO) corpus, with sentences annotated with one of the aforementioned categories. Unfortunately, since sentences in our primary source of positive data are from a different context than the abstracts to be classified, section headings, preceding sentences, and location are not available for our task.

2.1 Semi-Supervised Learning

Semi-supervised learning has the potential to match the performance of supervised learning while requiring considerably less labelled data (Wang et al., 2008; Thomas et al., 2012; Liu et al., 2013). Soft labelling (e.g. aforementioned heuristics for using section headlines as labels) is sometimes subsumed under semi-supervised learning as Distant Supervision (Go et al., 2009; Vosoughi et al., 2015; Wallace et al., 2016). Label Propagation (Zhu and Ghahramani, 2002) and Label Spreading (Zhou et al., 2003) can be seen as largely unsupervised classification, using labelled data to initialise and control clustering. Likewise, Nigam et al. (2011) propose Naive Bayes (NB) based variants of the unsupervised Expectation-Maximisation (EM) algorithm for utilising unlabelled data in semi-supervised text classification.

2.1.1 PU Learning

PU Learning is a special case of semi-supervised learning where examples in the unlabelled set U are to be classified as positive (label 1) or negative (label 0), with only positive labelled data P initially available. Therefore, the PU Learning problem can be approximated by learning to discriminate P from U (Mordelet and Vert, 2014). For that, learning should favour false positive errors over false negatives, e.g. by using class-specific weights for error penalisation. Approaches include *one-class SVMs*, which approximate the support of the positive class and treat negative examples as outliers; *ranking methods*, which rank unlabelled examples by their decreasing similarity to the mean positive example; and *two-step heuristics*, which try to identify reliable negative examples in the unlabelled data to initialise semi-supervised learning. We consider the aforementioned heuristics useful for outlier detection to reduce noise in our auxiliary data, and use variations of PU Learning algorithms in semi-supervised learning, as our problem of finding summary-like sentences without explicitly defined negative sentences is closely related to PU Learning. An overview is available in (Liu et al., 2003). Additional information can be found in (Elkan and Noto, 2008; Plessis et al., 2014, 2015). An example of the use of PU Learning, for spotting online fake reviews, is available in (Li et al., 2014).

Without known negative examples, measuring classification performance using accuracy or F1-score in PU Learning is not possible. Lee and Liu (2003) suggest an alternative score, called PU-score, defined as $Pr[f(X) = 1|Y = 1]^2/Pr[f(X) = 1]$, for comparing PU Learning classifiers that can be derived from positive and unlabelled data alone. The authors show theoretically that maximising the PU-score is equivalent to maximising the F1-score and can be used to compare different models classifying the same data. Nonetheless, it should be noted that this metric is not bounded, making it viable only for comparing classifiers trained and tested on the same data; it is not an indicator for an individual classifier's performance.

2.1.2 Self-Training

Self-Training, used in this work, starts from an initial classifier trained on the labelled data. Previously unlabelled examples that were labelled with high confidence are added to the training data. This procedure repeats iteratively, retraining the classifier until a terminating condition is met. NB is a popular classifier for Self-Training because the probabilities it produces provide a confidence ranking, but any other algorithm may be used as long as confidence scores can be derived (Wang et al., 2008).

3 Methods

We present the data sources we use, the preprocessing pipeline and describe in detail the experiments performed with both PU Learning and Self-Training.

3.1 Used Corpora

CIViC is a database of clinical evidence summaries. Entries consist of evidence statements about gene variants, such as their association with diseases and the outcome of drug response trials. Additional information includes the names of the respective genes, variants, drugs, diseases, and a variant summary. Each entry contains the PubMed

ID of the respective publication the information is taken from. Evidence statements are prototypes of high-quality information in condensed form. However, they are not themselves contained in the abstracts they are based on, as they try to summarize the entire article. At time of writing, CIViC contains about 2,300 evidence statements consisting of 6,600 sentences (5,300 without duplicates). They make up our initial corpus of positive sentences (P).

PubMed abstracts referenced in CIViC. We extract about 12,700 sentences from 1,300 abstracts referenced in CIViC, and use them as the unlabelled corpus (U). We use CIViC summaries and the PubMed abstract corpus to estimate the acceptable range for the ratio of key sentences in an abstract. We use the ratio of overall sentence counts in the two corpora (CIViC summaries and PubMed abstracts) as an upper bound of $\approx$ 0.4 (5,300/12,700). As a rough estimate for the lower bound, based on an informed guess that half of the sentences could be redundant, since one abstract may correspond to multiple CIViC entries for different drug/variant combinations. This results in a lower bound $\approx$ 0.2. Although this is a simplifying assumption and disregards e.g. any differences in information density in our data sources' sentences, it provides a rough guideline for the ratio of key sentences in U a classifier should find.

Hallmarks of Cancer (HoC) describe common traits of all forms of cancer. We use it as a silver standard corpus consisting of about 13,000 sentences from 1,580 PubMed abstracts. Sentences not relevant to any of the hallmarks are left unlabelled. We assume unlabelled sentences are less likely to be clinically relevant than sentences with one or more labels, aggregating them in the likely negative set HoC_n (about 8,900 sentences) and the likely positive set HoC_p (about 4,300 sentences). In order to improve generalisation, as well as to be able to validate our classifier, which requires positive as well as negative labelled data, we use HoC as auxiliary data. To utilise HoC_p and HoC_n as sources of realistic positive and negative sentences for training and test data, but avoiding propagation of misclassification errors resulting from our simplifying assumption, they must be filtered for noise (Section 3.3).

3.2 Text Preprocessing and Feature Selection

As features, we use word n-grams, character n-grams, and sentence length, concatenating them to form a mixed feature space. All tokens are converted to lower-case. Biomedical scientific text exhibits some particularities that have to be taken into consideration during text preprocessing. To normalise all text, before sentence splitting with the PunktSentenceTokenizer of the Python Natural Language Toolkit (NLTK) (Bird et al., 2009), we use regular expressions: we substitute spaces for full stops in common abbreviations followed by a space and lower-case letter or digit (e.g. "ca._5" → "ca__5"). As the pattern "patient no. V[123]" is quite frequent in *CIViC*, we introduce a special rule for not splitting it despite the upper-case. All whitespace characters are replaced by spaces to avoid splitting on newlines. Afterwards, to avoid character encoding-related problems and to reduce alphabet size, we normalize all text to ASCII before tokenization.

For word-level tokenization, we use NLTK's TreebankWordTokenizer and split the resulting tokens at characters in {"-", "/", ".", ",", "—", "¡", "¿"}. Sentences below a minimum character count of 8 are denoted by a special "_empty_sentence_" token. To prepare word n-grams, we replace tokens representing numbers or ordinals and their spelled out versions by a special "_num_" token and do the equivalent for e.g. ranges, inequalities, percentages, measurement units, and years. Tokens with suffixes common for drugs but not found in common speech, such as "-inib", are replaced by "_chemical_", and sequences that start with a letter, but contain digits, are replaced by "_abbrev_" in the hope of catching identifiers of biomedical entities. We evaluated the use of word n-grams with n bounded from (1,1) (the bag-of-words case) up to (1,4), thereby retaining information about the order of words in a sentence.

We evaluated the use of character n-gram with n in ranges from (2,3) to (2,6). To reduce alphabet size and avoid overfitting, all sequences of non-word characters except those in {"-", "%", "="}, which may carry semantic information, are replaced by single spaces, and all digits are replaced by 1.

In feature vectors, word and character n-grams are weighted by their tf-idf score (Aizawa, 2003) and sentence length is represented as inverse character count. Character n-grams proved to be more expressive than word n-grams, yielding better accuracy scores when we tested each of them in isolation. However, a combination of character and word level n-grams and text length performed best.

3.3 Noise reduction with PU Learning

Using PU Learning, we filter HoC_n and HoC_p for sentences that are likely to be useful in our classification task. We explored several approaches to PU Learning and subsequent noise reduction.

3.3.1 PU Learning

First, we explored several **Two-Step** techniques, which (1) identify a set of reliable negative examples (RN) from U and (2) train a classifier on P and RN and retrain the classifier using the predicted labels until a stopping criterion is met. We present them next. ***i-EM*** is a variation of the Expectation-Maximisation algorithm that relies on a NB classifier, with predicted probabilities of labels in range $[0, 1]$. ***s-EM*** is an extension to i-EM. Initially, a subset $S \subset P$ is added to U as spy documents. After training an initial classifier on $P \setminus S$ vs. $U \cup S$, the predicted probabilistic labels for the known hidden positives in S are used to determine a threshold t; all $u \in U$ with probabilistic labels $p(y_u) < t$ are moved to RN. In Step 2, starting from a classifier trained to discriminate P from RN, the EM algorithm is iterated as in i-EM until it converges or the estimated classification error is deteriorating. ***Roc-SVM*** uses the Rocchio algorithm for Step 1: Firstly, prototype vectors $\bar{p}$ for P and $\bar{u}$ for U are computed as a weighted differences between the two sets' respective average examples. Using these vectors, RN is defined as $\{u \in U : \cos(u, \bar{u}) < \cos(u, \bar{p})\}$, i.e. all unlabelled sentences that are more similar to the prototype of the unlabelled set than the positive sets. Step 2 uses SVMs to expand RN. Initially, an SVM is trained to discriminate P from RN. Afterwards, all $u \in U \setminus RN$ with predicted label 0 are added to RN for iteratively retraining the classifier as long as RN changes. This iteration may go wrong and result in poor recall on the positive class; as a fallback strategy, if the classifier at convergence misclassifies too large a portion of P, the initial classifier is returned instead. ***CR-SVM*** is a minor extension to Roc-SVM. P and U are each ranked by decreasing cosine similarity to the mean positive example; a probably negative set PN is built from the $u \in U$ with a lower score than a given ratio of least typical examples in P. The negative prototype vector is then computed using PN rather than U. Step 2 is the same as in Roc-SVM. Additionally, we explored **Biased SVM**, a soft-margin SVM that uses class-specific weights for positive and negative errors. Weight parameters are selected in a grid search manner to find a combination that optimises the PU-score; this effectively assumes U to contain only negligible amounts of hidden positive examples.

3.3.2 Noise reduction

We experiment with two heuristics for noise reduction in HoC. For both of them, let clf$(P, U)(x)$ be the label for x predicted by classifier clf trained on P and U. Appendix B (Figure 1) summarises corpora used for this task.

Strict mode: Remove *CIViC*-like sentences, i.e. likely hidden positives, from HoC_n for the reliable negative set HoC'_n. Keep only *CIViC*-like sentences in HoC_p for a reliable positive set HoC'_p. This implies rather pessimistic assumptions about HoC_p's relevance, considering only outliers as key sentences.

$$HoC'_n := HoC_n \setminus \{x \in HoC_n : \text{clf}(CIViC, HoC_n)(x) = 1\}$$
$$HoC'_p := \{x \in HoC_p : \text{clf}(CIViC, HoC_p)(x) = 1\}$$

Tolerant mode: Remove *CIViC*-like sentences from HoC_n as before. But rather than requiring sentences from HoC_p to be *CIViC*-like, remove those sentences from HoC_p that are similar to reliable negative sentences, i.e. the purified HoC'_n. In doing so, HoC_p is assumed to be largely relevant, contaminated with non-key sentences.

$$HoC'_n := HoC_n \setminus \{x \in HoC_n : \text{clf}(CIViC, HoC_n)(x) = 1\}$$
$$HoC'_p := HoC_p \setminus \{x \in HoC_p : \text{clf}(HoC'_n, HoC_p)(x) = 1\}$$

3.4 Semi-supervised learning with Self-Training

In the following, let the labelled set be $L := P \cup N$, with positive labelled set $P := CIViC \cup HoC'_p$ and negative labelled set $N := HoC'_n$. The unlabelled set of original abstracts is denoted by U. The purified sets HoC'_p and HoC'_n are obtained using either of the above heuristics. Appendix B (Figure 2) summarises corpora used for this task. We use: ***Standard Self-Training (ST)*** with a confidence threshold. Having experimented with different values, we use a threshold

	Reliable negatives: $P := CIViC$, $U := HoC_n$		Strict mode: $P := CIViC$, $U := HoC_p$		Tolerant mode: $P := HoC'_n$ $U := HoC_p$	
Method	PU-score	pos. ratio in U_{test}	PU-score	pos. ratio in U_{test}	PU-score	pos. ratio in U_{test}
i-EM	2.06	0.11	1.61	0.06	0.94	0.36
s-EM	2.06	0.11	1.61	0.06	0.94	0.40
Roc-SVM	**2.19**	**0.07**	**1.67**	**0.06**	**1.07**	**0.31**
CR-SVM	2.19	0.08	1.67	0.06	1.04	0.57
Biased-SVM	2.28	0.03	1.70	0.05	1.13	0.31

Table 1: Removing noise from HoC_n and HoC_p. Results for different PU Learning techniques, averaged over 10 runs, on 20% reserved test sets $P_{test} \subset P$ and $U_{test} \subset U$. To generate HoC'_n as required for tolerant mode, Roc-SVM (highlighted in bold) was used in the previous step.

Algorithm 1 Self-Training

```
procedure SELF-TRAINING(training data $L$ with labels, unlabelled data $U$)
    while $U$ is not empty do
        train classifier clf on $L$
        predict labels for $U$ with clf
        move examples with most confidently predicted labels from $U$ to $L$
    end while
    return clf
end procedure
```

of 0.75 for classifiers producing class probabilities, and 0.5 for the absolute values of SVM's decision function; ***"Negative" Self-Training (NST)***: Rather than using a confidence criterion, all unlabelled examples classified as negative are added to the training data for the next iteration. This is analogous to the iterative SVM step of Roc-SVM, except for the predefined rather than heuristically estimated initial negative set, and has shown to help avoid an unrestricted propagation of positive labels; A variant of the ***Expectation-Maximisation (EM)*** algorithm as used in i-EM. Starting with P and N as initial fixed-label examples, iterate a NB classifier until convergence, using the class probabilities predicted for U as labels for the next training iteration; ***Label Propagation and Label Spreading***: These algorithms propagate labels through high-density regions using a graph representation of the data. Both are implemented in Scikit-learn with Radial Basis Function (RBF) and k-Nearest-Neighbour (kNN) kernels available. We were unable to obtain competitive results with these techniques. In the Self-Training algorithm (shown in Algorithm 1), we use Scikit-learn's implementations of SVM, NB, and Logistic Regression (LR) as underlying classifiers.

4 Results

Section 4.1 describes the effects of noise reduction heuristics using PU Learning. The performances of different semi-supervised approaches for training a classifier, with both strict and tolerant noise reduction scenarios, are shown in Section 4.2.

4.1 PU Learning for Noisy Data

Table 1 summarises the PU-scores and ratio of examples in U classified as positive for different algorithms for reducing noise in HoC_n and HoC_p using the strict vs. tolerant heuristics.

Cleaning up HoC_n removes some 2 to 7% of examples, depending on the classifier. Additional manual inspection of a subset of the sentences removed confirms them as true negatives with respect to key sentences.

Regarding HoC_p, the strict heuristics keeps only 5.5%, some 250 sentences, of positive examples. We suspect this is due to the different thematic foci of HoC and the articles summarised in $CIViC$, as well as the summaries' different writing style. This leaves us with $N :=$ 8,300 sentences, $P :=$ 5,600 sentences and $U :=$ 12,700 sentences. As our experiments show, choosing this very selective approach drastically improves the nominal accuracy of subsequent steps; however, it leaves a lack of real-world data in the positive training set and harbours the risk of overfitting.

On the other hand, using the tolerant strategy, roughly 25% of HoC_p are removed due to being very similar to HoC'_n. This results in a 50%

			P_{test} :			N_{test} :			U :
Method	parameters	acc	p	r	F1	p	r	F1	pos. ratio
NST(SVM)	$C = 0.3$	0.94	0.95	0.90	0.92	0.93	0.97	0.95	0.33
NST(LR)	$C = 6.0$	0.89	0.99	0.75	0.84	0.86	0.99	0.92	0.13
NST(NB)	$\alpha = 0.1$	0.90	0.97	0.78	0.86	0.87	0.98	0.92	0.31
ST(SVM)	$C = 0.4$	0.96	0.97	0.93	0.95	0.95	0.98	0.97	0.62
ST(LR)	$C = 4.0$	0.96	0.96	0.94	0.95	0.96	0.98	0.97	0.62
ST(NB)	$\alpha = 0.1$	0.92	0.93	0.88	0.90	0.92	0.96	0.94	0.60
EM	$\alpha = 0.1$	0.91	0.92	0.85	0.88	0.91	0.95	0.93	0.62
Label Propagation	RBF kernel	0.83	0.91	0.63	0.74	0.79	0.96	0.87	0.35
Label Propagation	kNN kernel	0.69	0.96	0.22	0.36	0.66	0.99	0.79	0.03
Label Spreading	RBF kernel	0.85	0.93	0.68	0.79	0.82	0.97	0.89	0.50
Label Spreading	kNN kernel	0.79	0.92	0.54	0.68	0.76	0.96	0.85	0.32

Table 2: Performance of different semi-supervised approaches trained on P, N, and U after strict noise filtering. ST = Self-Training. NST = “Negative” Self-Training. Results averaged over 10 runs with randomised 20% validation sets from P and N; min-df threshold = 0.002, 25% of most relevant features selected with χ^2.

			P_{test} :			N_{test} :			U :
Method	parameters	acc	p	r	F1	p	r	F1	pos. ratio
NST(SVM)	***C* = 0.3**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.83**	**0.83**	**0.32**
NST(LR)	$C = 6.0$	0.81	0.88	0.72	0.79	0.76	0.90	0.82	0.17
NST(NB)	$\alpha = 0.1$	0.76	0.85	0.64	0.73	0.70	0.88	0.78	0.30
ST(SVM)	$C = 0.4$	0.85	0.90	0.81	0.85	0.83	0.89	0.86	0.62
ST(LR)	$C = 6.0$	0.86	0.87	0.85	0.85	0.84	0.86	0.85	0.66
ST(NB)	$\alpha = 0.1$	0.76	0.88	0.62	0.72	0.69	0.91	0.79	0.70
EM	$\alpha = 0.1$	0.74	0.88	0.58	0.70	0.68	0.91	0.78	0.70
Label Propagation	RBF kernel	0.72	0.88	0.50	0.64	0.64	0.92	0.76	0.36
Label Propagation	kNN kernel	0.58	0.90	0.20	0.32	0.54	0.98	0.70	0.02
Label Spreading	RBF kernel	0.74	0.88	0.56	0.68	0.67	0.92	0.77	0.56
Label Spreading	kNN kernel	0.68	0.91	0.43	0.58	0.62	0.96	0.77	0.34

Table 3: Performance of different semi-supervised approaches trained on P, N, and U after tolerant noise filtering. Results averaged over 10 runs with randomised 20% validation sets from P and N; min-df threshold = 0.002, 25% of most relevant features selected with χ^2. The model we consider most suitable for identifying key sentences is highlighted in bold.

larger and less homogenous positive labelled set compared to strict noise filtering, which we expect to provide greater generality and robustness to our classifier. This leaves us with N := 8,300 sentences, P := 8,600 sentences and U := 12,700 sentences. This is enough to assume a noticeable reduction of noise and easier distinction between HoC'_n and HoC'_p, but it still contributes a considerable amount of data to the positive set and is not suspect to overfitting. Typical topics of sentences removed as irrelevant include biochemical research hypotheses and non-human study subjects; however, as this heuristic is indirectly defined, its decisions are not quite as clearly correct as those directly linked to *CIViC*.

Our results confirm Biased-SVM nominally performs best among the PU Learning techniques described above; this is simply because the PU-score is maximised by minimising the amount of positive examples found in U, which Biased-SVM does by regarding U as negative and performing supervised classification. However, we do not find this to be useful for our purpose of noise detection, or for finding hidden positive data in unlabelled data in general. The EM-based techniques tend to go the opposite direction and consider comparably large ratios of U as positive, were more sensitive to distributions, and misclassified positive labelled data. Roc-SVM, on the other hand, had stable performance in our tests and scores close to those of Biased-SVM, which is why we use this approach to filter HoC for the subsequent steps. Our results also suggest the iterative second step is more crucial than the exact heuristics for choosing a reliable negative set from the unknown data.

4.2 Semi-Supervised classification of key sentences with Self-Training

We report accuracy (acc), precision (p), recall (r), F1-score ($F1$) and the ratio of key sentences found in U ($pos.ratio$) of different semi-supervised learning methods for strict (Table 2) and tolerant (Table 3) noise filtering scenarios. We consider classification to have gone wrong if the ratio of positive sentences in U significantly deviates from the acceptable range $[0.2, 0.4]$ (as defined in Section 3.1). Additionally, results of supervised ML pipeline on data sets generated after noise filtering are available in Appendix A.

Our experiments show that strict noise filtering leads to greatly improved classification accuracy; however, it may be fallacious to judge this approach only by the scores it produces. Given *CIViC*'s deviations from typical language in scientific articles, the different thematic foci of *CIViC* and *HoC*, and the negligible amount of realistic positive sentences added in this scenario (Table 1), we suspect classifiers may overfit to superficial and incidental differences rather than learning to generalise to correctly identify key sentences in unseen abstracts. In order to avoid this, we discard strict noise filtering.

On the other hand, tolerant filtering of HoC_n and HoC_p still allows for reasonable classification accuracy considering the data's heterogeneity. We expect additional positive sentences to provide improvements to generalisation that outweigh the lower nominal performance scores and possible errors propagated due to remaining noise. Although HoC's notion of relevant sentences is not identical to that implied by *CIViC*, our experiments show that removing only the least suitable sentences is enough to use HoC_p' as meaningful training data.

Standard Self-Training yields performance results very similar to supervised classification, analogous to what can be observed in strict mode, but a larger ratio of positive predictions for U. The linear classifiers SVM and Logistic Regression perform much better than NB, the latter modelling an inaccurate probability distribution. In both strict and tolerant mode, methods with an emphasis on unsupervised clustering (EM, Label Propagation, and Label Spreading) underperform, with a strong bias towards the negative class. Label Propagation with k-Nearest-Neighbours kernel performs particularly poorly, failing to find any positive examples in the unlabelled set. In contrast, NST with base classifiers leads to positive ratios in U close to our preliminary estimate, as well as acceptable classification accuracy. SVM performs better than Logistic Regression and has balanced precision and recall for both classes, appearing the more robust choice.

5 Conclusion

We have developed a pipeline for identifying the most informative key sentences in oncological abstracts, judging sentences' clinical relevance implicitly by their similarity to clinical evidence summaries in the *CIViC* database. To account for deviations from typical content between professional summaries and sentences appearing in abstracts, we use the abstracts corresponding to these summaries as unlabelled data in an semi-supervised learning setting. An auxiliary silver standard corpus is used for more realistic training and validation data. To mitigate introducing errors due to miscategorised examples in partitions of the auxiliary data, we propose using PU Learning techniques in a noise detection preprocessing step.

We evaluate different heuristics for semi-supervised learning and measure their performance with heterogenous data. While methods with an emphasis on unsupervised clustering perform poorly, (which we attribute to the data violating smoothness assumptions) Self-Training with linear classifiers proved robust to unfavourably distributed data, reaching performance scores similar to those of supervised classifiers trained without the unlabelled data. By adapting Self-Training with SVMs to iteratively expand only the negative training set as in PU Learning, we were able to restrict the amount of hidden positive examples found in unlabelled data while maintaining good accuracy scores. Our best model using this method reaches 84% accuracy and 0.84 F1-score.

As a byproduct of the proposed pipeline, we obtain a silver standard corpus consisting of approximately 12,700 sentences from our unlabelled set, annotated with sentences' estimated clinical relevance, which may be useful for future classification tasks. Our final pipeline can be used to help clinicians quickly assess which articles are relevant to their work, e.g. by incorporating it into workflows for the retrieval of cancer-related literature. As such, it has been integrated in to Variant Information Search Tool[1] (VIST), a query-based document retrieval system which ranks scientific abstracts according to the clinical relevance of their content given a (set of) variations and/or genes.

We encountered various difficulties resulting from using a gold standard with atypical and solely positive examples and the heterogeneity of different training corpora. Although our problem of finding key sentences is a standard PU Learning task, the methods described in the PU Learning literature cannot be used in a verifiable way on real-world data without negative validation data. Even for semi-supervised learning with positive as well as negative labelled data, standard metrics alone are not enough to judge a classifier's adequacy, since the amount of noise in automatically gathered training data is never completely certain and the way unlabelled data is handled by a classifier is not represented in performance scores. By using heuristics for noise filtering and adapting self-training to incorporate unlabelled data in a way suitable to our goal, we alleviate these difficulties.

References

Shashank Agarwal and Hong Yu. 2009. Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, 25(23):3174–3180.

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. Neural networks for joint sentence classification in medical paper abstracts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 694–700. Association for Computational Linguistics.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Malachi Griffith, Nicholas C Spies, Kilannin Krysiak, Joshua F McMichael, Adam C Coffman, Arpad M Danos, Benjamin J Ainscough, Cody A Ramirez, Damian T Rieke, Lynzey Kujan, et al. 2017. Civic is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. *Nature genetics*, 49(2):170.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

A. Hassan and A. Mahmood. 2017. Deep learning for sentence classification. In *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–5.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98, Chemnitz, Germany*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer.

Anthony Khoo, Yuval Marom, and David Albrecht. 2006. Experiments with sentence classification. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 18–25.

Su Kim, David Martínez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. *BMC Bioinformatics*, 12(S-2):S5.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

[1] https://triage.informatik.hu-berlin.de:8080/

Julian Kupiec, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *SIGIR'95, Seattle, Washington, USA*, pages 68–73. ACM Press.

Wee Sun Lee and Bing Liu. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 448–455. AAAI Press.

Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. 2014. Spotting fake reviews via collective positive-unlabeled learning. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 899–904, Washington, DC, USA. IEEE Computer Society.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2003. Building text classifiers using positive and unlabeled examples. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), Melbourne, Florida, USA*, pages 179–188. IEEE Computer Society.

Zhiguang Liu, Xishuang Dong, Yi Guan, and Jinfeng Yang. 2013. Reserved self-training: A semi-supervised sentiment classification method for chinese microblogs. In *Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan*, pages 455–462. Asian Federation of Natural Language Processing / ACL.

Larry McKnight and Padmini Srinivasan. 2003. Categorization of sentence types in medical abstracts. In *American Medical Informatics Association Annual Symposium, Washington, DC, USA*. AMIA.

Fantine Mordelet and Jean-Philippe Vert. 2014. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209.

National Library of Medicine. 1946-2018. Pubmed. `https://www.ncbi.nlm.nih.gov/pubmed`. Accessed: 2018-02-01.

Bhawna Nigam, Poorvi Ahirwal, Sonal Salve, and Swati Vamney. 2011. Document classification using expectation maximization with semi supervised learning. *CoRR*, abs/1112.2028.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 703–711, Cambridge, MA, USA. MIT Press.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2015. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1386–1394, Lille, France. PMLR.

Anthony Rios and Ramakanth Kavuluru. 2015. Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '15, pages 258–267, New York, NY, USA. ACM.

Patrick Ruch, Célia Boyer, Christine Chichester, Imad Tbahriti, Antoine Geissbühler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, and Anne-Lise Veuthey. 2007. Using argumentation to extract key sentences from biomedical abstracts. *I. J. Medical Informatics*, 76(2-3):195–200.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Philippe Thomas, Tamara Bobić, Ulf Leser, Martin Hofmann-Apitius, and Roman Klinger. 2012. Weakly labeled corpora as silver standard for drug-drug and protein-protein interaction. In *Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM) on Language Resources and Evaluation Conference (LREC)*.

Soroush Vosoughi, Helen Zhou, and Deb Roy. 2015. Enhanced Twitter Sentiment Classification Using Contextual Information. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 16–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Byron C. Wallace, Joël Kuiper, Aakash Sharma, Mingxi (Brian) Zhu, and Iain James Marshall. 2016. Extracting PICO sentences from clinical trial reports using supervised distant supervision. *Journal of Machine Learning Research*, 17:132:1–132:25.

Bin Wang, Bruce Spencer, Charles X. Ling, and Harry Zhang. 2008. Semi-supervised self-training for sentence subjectivity classification. In *Advances in Artificial Intelligence , 21st Conference of the Canadian Society for Computational Studies of Intelligence Windsor, Canada*, pages 344–355. Springer.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In *NAACL HLT 2016, San Diego California, USA*, pages 1522–1527. The Association for Computational Linguistics.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A c-lstm neural network for text classification. *CoRR*, abs/1511.08630.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 321–328. MIT Press.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. pages 3485–3495.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation.

A Semi-supervised ML on filtered datasets

Table 4 shows results of supervised baseline classifiers trained on P and N after strict filtering. Performance is very good for all classifiers tested, which is not surprising as *CIViC* and HoC_n are easy to separate even without filtering. The ratio of the unlabelled set U classified as positive, however, is outside of the acceptable range [0.2, 0.4] for selecting key sentences, probably due to the more similar contents of *CIViC* and the corresponding abstracts compared to HoC.

Table 5 shows the results of supervised classifiers trained on only P and N after tolerant filtering. Accuracies and F1-scores are about 10 percent points lower compared to results in the strict filtering scenario, which can be explained by HoC_p and HoC_n being comparably difficult to separate. However, performance is better compared to distinguishing $CIViC \cup HoC_p$ vs. HoC_n without any noise filtering.

B PU Learning and Self-Training: used corpora

			P_{test} :			N_{test} :			U:
Method	parameters	acc	p	r	F1	p	r	F1	pos. ratio
SVM	$C = 3.0$	0.96	0.97	0.94	0.95	0.96	0.98	0.97	0.63
LR	$C = 6.0$	0.96	0.96	0.94	0.95	0.96	0.97	0.97	0.64
NB	$\alpha = 0.1$	0.94	0.95	0.91	0.93	0.94	0.97	0.95	0.61

Table 4: Supervised classifiers trained on P and N after strict noise filtering. Results averaged over 10 runs with randomised 20% reserved test sets; min-df threshold = 0.002, 25% of most relevant features selected with χ^2.

			P_{test} :			N_{test} :			U :
Method	parameters	acc	p	r	F1	p	r	F1	pos. ratio
SVM	$C = 3.0$	0.86	0.88	0.84	0.86	0.84	0.88	0.86	0.63
LR	$C = 6.0$	0.86	0.88	0.85	0.86	0.85	0.87	0.86	0.63
NB	$\alpha = 0.1$	0.79	0.89	0.67	0.77	0.72	0.91	0.81	0.70

Table 5: Supervised classifiers trained on P and N after tolerant noise filtering. Results averaged over 10 runs with randomised 20% validation sets; min-df threshold = 0.002, 25% of most relevant features selected with χ^2.

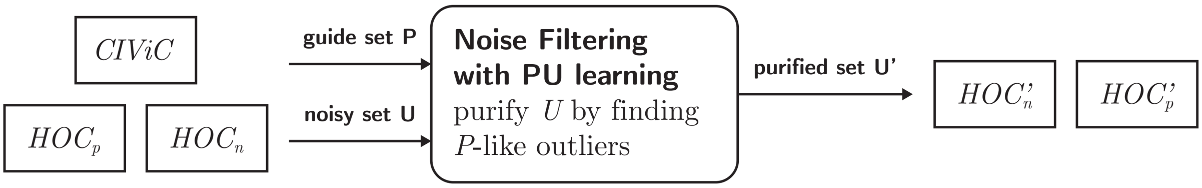

Figure 1: PU Learning for noise reduction - used corpora

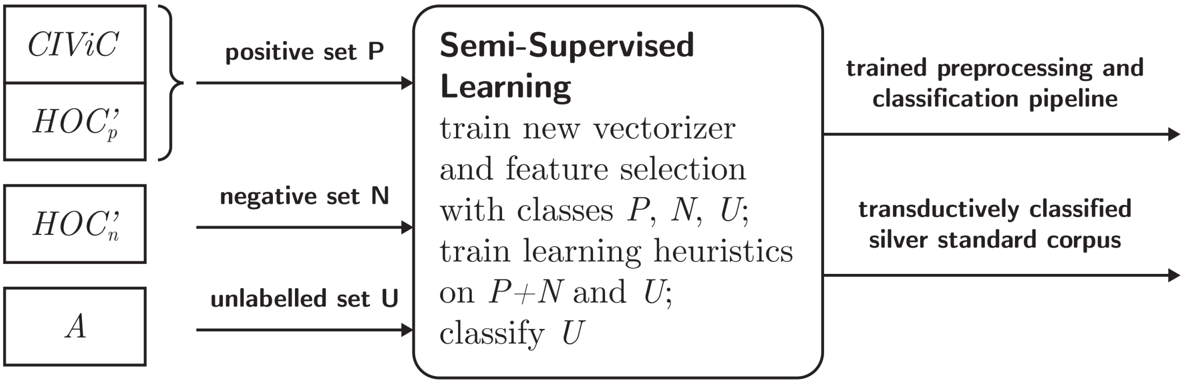
CIViC
HOC'_p
HOC'_n
A
positive set P
negative set N
unlabelled set U
Semi-Supervised Learning
train new vectorizer and feature selection with classes P, N, U; train learning heuristics on P+N and U; classify U
trained preprocessing and classification pipeline
transductively classified silver standard corpus

Figure 2: Semi-supervised training with Self-Training - used corpora