Using Dependency Relations for Text Classification

Vivi Nastase
EML Research gGmbH
Heidelberg, Germany
nastase@eml-r.org

Jelber Sayyad Shirabad
SITE
University of Ottawa
Ottawa, ON, Canada
jsayyad@site.uottawa.ca

Maria Fernanda Caropreso
SITE
University of Ottawa
Ottawa, ON, Canada
caropres@site.uottawa.ca

Abstract

We investigate the use of syntactically related pairs of words for the task of text classification. The set of all pairs of syntactically related words should intuitively provide a better description of what a document is about, than the set of proximity-based N-grams or selective syntactic phrases. We generate syntactically related word pairs using a dependency parser. We experimented with Support Vector Machines and Decision Tree learners on the 10 most frequent classes from the Reuters-21578 corpus. Results show that syntactically related pairs of words produce better results in terms of accuracy and precision when used alone or combined with unigrams, compared to unigrams alone.

1 Introduction

Text classification is an active field of research in machine learning and is applied to domains as diverse as spam detection [Drucker *et al.*, 1999] and sentence selection for bioinformatics [Nedellec *et al.*, 2003]. The most common representation method in this task is *vector representation* or *bag of words* [Sebastiani, 2002]. Values of features in this vector can be measures that summarize certain information about the words appearing in a corpus, such as TF/IDF [Furnkranz *et al.*, 1998], or they can be binary, indicating whether or not a word is present in a document.

The motivation for finding alternative features is the fact that words by themselves cannot capture the gist of a document. Several statistical and Natural Language Processing (NLP) inspired methods have been researched in the past. We propose an alternative, consisting of syntactically related pairs, generated using a dependency parser. Intuitively, the set of all syntactically related pairs should capture better what the text is about than N-grams[1], also called statistical phrases in the Machine Learning (ML) field, or syntactic phrases[2] that have been explored until now.

Syntactically related pairs are better than N-grams at capturing related concepts, and therefore less likely to introduce noise[3]. It is an accepted view within NLP community that syntactically related words and

[1] N-grams are proximity-based sequences of words, obtained by sliding a window of size N over the text. Stop words may have previously been removed.

[2] Syntactic phrases are usually constructed by selecting phrases that follow specific structures according to grammatical rules.

[3] Noise from a language point of view – from the bigrams generated for the phrase “black large bear”, the bigram “black large” is noisy, as the two words are not syntactically/semantically connected, as opposed to the word pair captured in the bigram “large bear”.

phrases are expressions of semantically related concepts. This view is reflected for example in the fact that among the first semantic relations researched were those between a verb and its arguments – which are of course syntactically related [Fillmore, 1968]. By using syntactically related words we are thus one step closer to a more semantic view of a document. Similar to N-grams, such pairs will cover the entire document. The type of syntactic phrases most commonly used for text classification follow specific syntactic patterns. The most commonly used are noun-phrases, as we will see in Section 2. When selecting specific syntactic structures from a document, a large number of unwanted structures are discarded. The omitted structures may contain relevant document information. Syntactically related word pairs of the type we extract provide a more exhaustive text description than syntactic phrases.

We use an off-the-shelf dependency parser to obtain syntactically related word pairs from the Reuters-21578 document collection. Dependency parsers have been in use for quite a while. MiniPar [Lin, 1998] is one of the free dependency parsers in use. Dependency parsers find pairs of syntactically connected words in a sentence. Building parse trees is possible, but optional, with such parsers. The advantages of using a dependency parser are that we easily obtain pairs of syntactically related words.

In this paper we investigate whether this shallow semantic approach improves text classification results, compared to the classic bag of words and bigrams approach. We discuss how one can use fast and scalable dependency parsing to build a new representation for documents in a text classification task. We applied these techniques to the Reuters-21578 text categorization collection[4]. We experimented with classifying the 10 most frequent classes in the collection. We reproduce experiments with the bag of words approach – unigrams and lemmatized unigrams – to provide a comparison baseline for dependency pair experiments. We observe that representing documents in the Reuters collection using syntactically related word pairs gives better results than the simple bag of words approach. We experiment with 8 sets of features and two machine learning algorithms.

In Section 2 we provide some of the related research in this area. Section 3 is dedicated to details of feature construction techniques we have employed. In Section 4 we present the experimental setup and discuss the results. Detailed analysis of the data and the impact of the dependency relation representation on the results obtained are discussed in Section 5. Our conclusions and future research direction can be found in Section 6.

2 Related Work

The bag of words (BOW) approach provides a very simple and easy to build language model. Less than perfect scores for text classification based on such a model show that BOW does not fully capture the gist of a document. It is natural then that researchers are searching for alternative or complementary representation methods. Inspiration has come from the field of statistics and NLP. N-grams and syntactic phrases (mostly noun-phrases) have been intensively investigated for the task of automated text classification (ATC) and information retrieval (IR). More intensive knowledge based methods have also been tried (see [Jacobs, 1992] for examples), but such methods are slower, and do not scale up with the size of the collection. The overall usefulness of noun-phrases and statistical phrases for text classification is still under debate.

Statistic phrases were shown to be better than noun phrases for IR [Fagan, 1987], while also being easier to obtain. For both ATC and IR, Lewis and Croft [1990], Lewis [1992b] and Lewis [1992a] have not

[4] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

observed an improvement when noun phrases were used instead of words. Mitra *et al.* [1997] show that using noun phrases leads to considerable improvement when words perform poorly, but they are not useful when words alone perform well.

Furnkranz *et al.* [1998] study the use of linguistic phrases with particular syntactic patterns for the text classification of web pages. These patterns are learned from the data by an extraction system. The results show these features can improve the precision of the classifiers at the low recall end. Dumais *et al.* [1998] and Scott and Matwin [1999] did not observe a significant improvement in classification on the Reuters-21578 collection, when noun-phrases were used. Statistical phrases improve text classification in some cases, according to Furnkranz [1998], Mladenic and Grobelnik [1998] and Caropreso *et al.* [2001]. Caropreso *et al.* [2001] experiment with N-grams (unigrams and bigrams) for text classification. An N-gram is considered to be a sequence of alphabetically ordered sequence of N consecutive words in a sentence, after stop words are removed. Analysis of unigrams and bigrams is performed using information gain, mutual information, χ^2 and other feature selection measures. While for these measures bigrams score sometimes higher than unigrams, in actual text classification experiments, the results (in terms of accuracy) do not improve significantly.

Dumais *et al.* [1998] test various classification techniques and feature selection methods on the Reuters-21578 data set (the ModApte split). The best results are obtained for SVM on a representation using a unigram-based feature set, selected using Mutual Information. They achieve a 92.0% BEP on the 10 most frequent categories of Reuters. This is the best classification result on the Reuter's 10 most frequent classes that we found in the literature.

Shapire and Singer [1998] and Weiss *et al.* [1999] use boosting algorithms with decision stumps and decision trees respectively to achieve 86% BEP on all Reuters categories, and 87.8% respectively, on the 95 largest categories (the ModApte split).

Structure of the categorized texts – document title, headings – has also provided features that are often more informative than features extracted from the body of a document [Furnkranz, 1999].

Within the field of NLP, we find similar work on finding ways to represent the topic of a text. Lin and Hovy [2000] present a first method to extract *topic signatures*, which are similar to the bag of word representations. The words are filtered based on frequency analysis in a set of relevant and a set of irrelevant documents. Harabagiu [2004] takes this one step further and uses relations extracted from texts - which are *(syntactic relation, word*$_1$*, word*$_2$*)* tuples – extracted using a syntactic parser, and based on the words from the (Lin and Hovy's) topic signatures. This analysis is used to find salient relation tuples for the expansion of questions in a question answering system [Harabagiu *et al.*, 2006], and the expansion of topics in a topic-driven summarization system [Harabagiu *et al.*, 2007].

The approach we present also makes use of the syntactic analysis of texts. We process texts using a dependency parser, extract all syntactically related word pairs, and select the most discriminating ones based on feature analysis. We use only features extracted from the body of the documents. The data sets obtained for the 8 feature sets generated are processed using SVM light[5] and C5.0[6].

[5] http://svmlight.joachim.org

[6] http://www.rulequest.com

3 Data representation

We test the usefulness of syntactically related word pairs to the task of text classification. Using freely available dependency parsers, syntactically related word pairs are easy to obtain, are less noisy than bigrams, and provide a more complete document description than statistic and syntactic phrases. We use the MiniPar dependency parser [Lin, 1998] to obtain a first approximation for our representation, which we then post-process as described in Section 3.1.

We compare the results obtained using two word-pair representations (one containing syntactic relation information between the words in a pair, the other not) with results obtained using unigrams and lemmatized unigrams, and pairwise combinations of these four feature sets.

In this section we describe in detail how the representations used were obtained.

A few considerations apply for all feature sets. All features used are binary, where a value of 1 means the feature was present in the document and 0 means it was not. We use the subset of 10 most frequent categories from the Reuters-21578 collection, with the ModApte split. The ModApte split designates specific documents for training and testing for each class in the collection. The data set on which the experiments are performed is obtained after several steps of filtering and processing, as follows:

1. Filter out documents that contain no text, or do not belong to one of the 10 most frequent classes;
2. Parse the body of the text with MiniPar, post-process, and collect word dependency pairs as features (no title or structural information is used);
3. Filter out features that do not appear in at least 2 documents;
4. Filter out documents that have an all 0 representation vector.

From the final set of documents and their corresponding features we generate the 8 datasets we experiment with. The training and testing sets are processed separately. The feature sets are derived from the analysis of the training corpus and then used to generate both the training and testing sets. The testing sets are not filtered. Table 1 shows the distribution of the positive class in the training and testing sets for each of the 10 classes considered. There are 5912 examples in the training set after filtering, and 2312 examples in the testing set.

3.1 Syntactically related word pairs representation

To be able to run MiniPar on the Reuters data collection, we split each document into individual sentences, using end of sentence punctuation and heuristics to avoid interpreting abbreviations as sentence terminators. The sentences are put one per line, and the file containing all sentences is read and processed by MiniPar using the lemmatizing and dependency pair generating options.

A dependency pair is a pair of grammatically related words: the main verbs in two connected clauses, a verb and each of its arguments, a noun and each of its modifiers. Some particularities of dependency grammars make necessary a post-processing step. We exemplify this thourgh a sample parse generated by MiniPar for the sentence:

Paris is the capital of France.

Class	Training set		Test set	
	examples	(% positive)	examples	(% positive)
acq	1489	(25.17%)	644	(27.85%)
corn	160	(2.70%)	48	(2.08%)
crude	353	(5.97%)	164	(7.09%)
earn	2692	(45.50%)	1036	(44.8%)
grain	399	(6.74%)	134	(5.80%)
interest	290	(4.90%)	100	(4.33%)
money-fx	462	(7.81%)	141	(6.10%)
ship	194	(3.28%)	87	(3.76%)
trade	339	(5.73%)	113	(4.89%)
wheat	199	(3.36%)	66	(2.85%)

Table 1: Positive class distributions

fin	C:i:VBE	be
be	VBE:s:N	Paris
be	VBE:pred:N	capital
capital	N:subj:N	Paris
capital	N:det:Det	the
capital	N:mod:Prep	of
of	Prep:pcomp-n:N	France

The parser output shows the dependency related words, their parts of speech, and the syntactic relation between them. *fin* is an internal symbol, which connects to the main verb of the sentence. It is interesting to notice that when the main verb of the sentence is *be*, MiniPar will consider the predicate to consist of *be* and the verb complement, and it will connect the subject with the complement, bypassing the verb (*capital N:subj:N Paris*). This is a good feature, as it generates the same pair when a modifier appears as the modifier of the noun, or as complement of the verb *be*. For example, the expressions *interesting paper* and *the paper is interesting* will result in the same pair *paper N:mod:Adj interesting*.

The above parse also shows why we need a post-processing step:

capital	N:mod:Prep	of
of	Prep:pcomp-n:N	France

First, we filter out pairs in which one of the elements is a MiniPar internal symbol, or a closed class word – for example, determiner (but not prepositions or coordinators, subordinators). Second, we compress two or more pairs through a "connective bypassing" process, as we show below, such that we obtain only pairs containing open-class words (nouns, verbs, adjectives and adverbs). In the example above, we combine the two tuples to produce the dependency *(of,capital,France)*. This type of compression is performed for pairs containing prepositions and clause subordinators and coordinators.

We generate two representations based on dependency pairs: one that contains information about the syntactic relation or connective between the words, and one that does not. The purpose is to verify if further compression can be obtained by disregarding the syntactic relation (as the same two words may co-occur in different syntactic configurations) and if this omission affects text classification results.

The processing described above produces a dependency pair feature set of 187836 elements when syntactic relations are present (we call it DRY – Dependencies & Relation Yes), and feature set of 170366

Class	DRN	DRY	U	UL
acq	7617	7430	1130	947
corn	4156	4024	1132	949
crude	4235	4025	1189	1050
earn	4314	4988	1132	911
grain	5843	5376	1130	914
interest	4525	4360	1130	912
money-fx	4745	4967	1321	936
ship	4172	4041	1130	914
trade	4208	4410	1130	910
wheat	4346	4035	1264	913

Table 2: Number of features for the 4 representations, per class

elements when the syntactic relation is omitted (DRN). After filtering from both sets features that do not appear in at least two documents, DRY will have 40159 elements and DRN 41431.

3.2 Unigram representation

The baseline representation to beat in document classification has been for quite a while the bag of words model. In order to compare the results of the dependency pair representation with the unigram model, there are two options: compare with results previously published in the literature, or perform the experiments anew. As we have shown in Section 3, in order to perform experiments using word dependencies we must perform several document filtering steps. This leads us to a different collection than used in other research, on which previous results have not been published. We are therefore forced to resort to the second alternative – reproduce bag of words experiments on the collection.

We create two bag of words representations: words/unigrams (U) as they appear in text – 19918 features – and unigrams lemmatized (UL) – 16461 features. After filtering out features that appear in only one document, we obtain a set of 11307 unigrams, and 9101 lemmatized unigrams.

We do not explicitly eliminate stop words. Instead, we filter out later in the process unigrams with low information gain score. This step filters out many of the closed class words (determiners, prepositions, pronouns, etc.) which are commonly part of the stop words list.

3.3 Generating datasets

The training and testing sets are represented using the four sets of features U, UL, DRY and DRN. Because the feature sets are large, they are filtered using information gain (IG)[Manning and Schütze, 1999].

The final feature sets consist of 10% of the features with the highest IG values. Because more than one feature may have an IG value equal to the cut-off point, the feature subsets may contain slightly more than the intended top 10% of features. Table 2 shows the number of features for each class and representation type. We create for each feature set 10 binary classification file sets (training and testing), corresponding to the 10 most frequent classes in the collection.

In order to investigate the potential of dependency relations not only as alternative, but also as complementary to unigrams, we generate four additional feature sets by combining each of the dependency relation (word pair) feature sets with each of the unigram-based sets (DRN-U, DRY-U, DRN-UL, DRY-UL). The training and testing data sets are generated by simple concatenation of the source feature vectors representing each example.

4 Experiments

We perform classification experiments for the 10 most frequent classes in the Reuters 21578 document collection, using 8 feature sets and two machine learning tools: C5.0 and SVM light. C5.0 was used with the default options, and SVM light used linear kernel, the other parameters set to their default values. The performance of the two ML tools was similar, as shown in Tables 3 – 6. SVM performed slightly better in terms of accuracy and precision.

Independent of the learning algorithm, the best average accuracy was obtained for the data represented using syntactically related pairs. Omitting the syntactic relation leads to a slight loss in accuracy. These two representations also have the best precision in classifying the positive class, although they also have the lowest recall. Lemmatized unigrams perform the worst in terms of accuracy, but when combined with word dependency pairs the performance improves, although not to the level of word pairs alone. In terms of per-class scores, a representation using dependency relations produced better F1-score results than unigrams and unigrams lemmatized. For 9 out of 10 classes, a representation using dependency relations produced equal or better accuracy results than unigrams and unigrams lemmatized.

It is interesting to notice that, as hypothesized, when we use dependency relations by themselves, both C5.0 and SVM light classifiers produce the results with the best precision, while unigrams and unigrams lemmatized have the best recall. This follows the intuition that, compared to words (unigrams), syntactically related word pairs produce a document topic representation which is one step closer to a semantic representation. Unigrams, on the other hand, have a broader coverage than word pairs do, so we are able to find more of the documents in the same topic, though at the cost of precision.

Combining feature sets brings up more interesting avenues to explore. While for most classes, a combination of unigrams and word pairs leads to better accuracy, there is no feature set that clearly outperforms others. There are several approaches to try at this point: feature selection on the combined sets – although filtered, the feature sets that represent the data are quite large, and in a combined set the size increases further; ansambles – find a way to combine classifiers to obtain a performance better than individual classifiers by themselves.

5 Data analysis

As mentioned in the introduction, the experiments with unigrams and lemmatized unigrams were performed to provide a comparison baseline for the performance of dependency pairs for the text classification task. The reason a published result could not be used for comparison is the data filtering process we had to perform (detailed in Section 3), which lead to a subset of the 10 most frequent classes from the Reuters-21578 collection different than what was used in previously published work.

We performed data analysis, to get a better understanding of the characteristics of the texts in the Reuters-21578 collection, and the impact of these characteristics on the results obtained.

Individual feature sets

Class	DRN		DRY		U		UL	
	P	R	P	R	P	R	P	R
acq	41.5	23.14	**56.77**	27.33	37.48	88.98	36.44	*91.77*
corn	1.65	8.33	**8.7**	8.33	0	0	3.1	**50**
crude	**25.64**	18.29	21.82	43.9	17	52.44	10.18	*72.56*
earn	15.25	6.76	15.02	3.67	**21.89**	**26.35**	12.33	5.31
grain	6.55	14.18	**31.19**	**25.37**	6.47	9.7	9.68	24.63
interest	15.87	20	**32.35**	22	5.68	**76**	7.79	74
money-fx	**47.97**	41.84	26.04	17.73	9.92	**71.63**	5.93	65.96
ship	1.98	4.6	**6.71**	11.49	6.33	33.33	4.07	**48.28**
trade	**12.24**	21.24	9.39	15.04	10.18	69.03	9.32	**80.53**
wheat	**11.63**	**15.15**	2.04	3.03	0	0	1.43	1.52
micro Avg.	17.68	15.36	**24.39**	15.79	17.38	**48.53**	12.55	44.33
macro Avg.	18.03	17.35	**21.00**	17.79	11.50	42.75	10.03	**51.46**

Combined feature sets

Class	U_DRN		U_DRY		UL_DRN		UL_DRY	
	P	R	P	R	P	R	P	R
acq	**41.35**	80.9	40.83	79.19	36.58	94.57	37.02	**95.03**
corn	0	0	0	0	**3.11**	**52.08**	**3.11**	**52.08**
crude	13.38	57.93	**13.76**	60.98	10.7	**61.59**	9.43	60.98
earn	**35.13**	**51.64**	22.63	28.28	12.48	6.95	13.49	11.68
grain	3.84	**40.3**	3.89	39.55	1.74	14.18	**7.61**	15.67
interest	8.28	28	7.57	**69**	10.65	33	**18.29**	30
money-fx	14.08	75.18	11.47	**82.98**	15.04	50.35	**15.16**	29.79
ship	**7.13**	36.78	6.19	31.03	3.93	**47.13**	3.84	42.53
trade	11.16	**69.03**	**16.16**	69.91	9.47	62.83	9.64	50.44
wheat	0	0	0	0	0	0	0	0
micro Avg.	**20.29**	**57.20**	16.65	49.27	13.60	41.14	15.62	41.26
macro Avg.	**13.44**	43.98	12.25	**46.09**	10.37	42.27	11.76	38.82

Table 3: Precision and recall obtained using C5.0

Class	DRN	DRY	U	UL	U_DRN	U_DRY	UL_DRN	UL_DRY
acq	**69.5**	74	55.6	53.1	62.7	62.2	52.8	53.6
corn	87.8	96.3	**97.7**	66.6	**97.7**	**97.7**	65.3	65.3
crude	**90.4**	84.9	78.5	52.6	70.4	70.1	60.8	55.7
earn	41.4	**47.5**	24.9	40.7	35.6	24.5	36.5	26.9
grain	83.3	**92.4**	86.6	82.3	38.1	39.8	48.6	84.1
interest	92	**94.6**	44.4	61	83.5	62.2	85.1	91.2
money-fx	**93.7**	91.9	58.6	34.1	70.5	59.9	79.6	85.6
ship	87.8	**90.7**	78.9	55.3	79.6	79.7	54.7	57.7
trade	88.7	**88.8**	68.7	60.8	71.6	80.8	68.8	74.5
wheat	94.3	93.1	**97.1**	94.2	**97.1**	**97.1**	97	97
micro Avg.	82.90	**85.40**	69.10	60.10	70.70	67.40	64.90	69.20
macro Avg.	82.89	**85.42**	69.10	60.07	70.68	67.40	64.92	69.16

Table 4: Accuracy obtained using C5.0

Individual feature sets

Class	DRN		DRY		U		UL	
	P	R	P	R	P	R	P	R
acq	**52.1**	9.63	48.66	**14.13**	8.91	2.8	25.75	13.35
corn	25	8.33	**33.33**	4.17	8.33	4.17	2.68	**35.42**
crude	71.79	17.07	**83.33**	12.2	25.35	43.9	17.83	**50**
earn	10.69	1.64	43.56	6.85	**48.89**	2.12	44.88	**8.88**
grain	25.25	18.66	**26.19**	16.42	12.39	**20.9**	6.13	7.46
interest	**66.67**	8	62.5	10	14.93	10	14.69	**31**
money-fx	62.71	26.24	**68.42**	18.44	26.37	**37.59**	16.85	32.62
ship	6.9	2.3	**11.11**	4.6	9.17	**82.76**	6.67	**82.76**
trade	**68**	15.04	66.67	12.39	29.46	33.63	24.01	**64.6**
wheat	13.04	**4.55**	9.09	1.52	**14.29**	1.52	11.11	1.52
micro Avg.	35	8.01	**44.54**	10.30	16.04	12.48	13.89	**20.13**
macro Avg.	40.21	11.14	**45.28**	10.07	19.81	23.94	17.06	**32.76**

Combined feature sets

Class	U_DRN		U_DRY		UL_DRN		UL_DRY	
	P	R	P	R	P	R	P	R
acq	**21.34**	**8.39**	19.48	8.07	5.31	0.93	4.46	0.78
corn	**3.05**	**20.83**	2.55	18.75	0	0	0	0
crude	23.51	**40.85**	22.86	39.02	**28.08**	34.76	27.92	33.54
earn	41.84	7.92	41.67	**8.2**	**47.92**	2.22	**47.92**	2.22
grain	8	7.46	6.06	5.97	13.64	11.19	**14.04**	**11.94**
interest	26.92	14	29.82	**17**	**31.25**	5	**31.25**	5
money-fx	33.03	25.53	32.04	23.4	45.05	**29.08**	**47.83**	23.4
ship	7.85	73.56	7.82	**77.01**	11.54	65.52	**12.01**	73.56
trade	**28.43**	**49.56**	28.42	47.79	27.42	15.04	26.67	17.7
wheat	0	0	0	0	**50**	**1.52**	25	**1.52**
micro Avg.	16.56	**15.52**	15.87	15.36	**19.34**	8.76	18.89	8.76
macro Avg.	19.39	**24.81**	19.07	24.52	**26.02**	16.52	23.71	16.96

Table 5: Precision and recall obtained using SVM Light

Class	DRN	DRY	U	UL	U_DRN	U_DRY	UL_DRN	UL_DRY
acq	**72.36**	71.93	64.97	65.14	67.78	67.73	65.87	65.1
corn	97.58	**97.84**	97.06	71.97	97.53	97.62	84.6	83.43
crude	**93.64**	93.6	86.85	80.1	89.06	89.14	86.38	86.33
earn	49.78	54.28	**55.15**	54.28	55.1	55.1	53.81	53.72
grain	92.08	**92.47**	86.85	88.02	90.74	90.66	89.66	89.19
interest	**95.85**	**95.85**	93.64	89.23	95.42	95.42	94.64	94.68
money-fx	**94.55**	94.51	89.79	86.07	93.51	93.77	92.3	92.3
ship	**95.16**	95.03	68.51	55.8	79.8	78.72	66.52	64.97
trade	**95.5**	95.42	92.82	88.28	93.9	93.6	91.44	91.57
wheat	96.41	**96.76**	96.93	96.84	97.15	97.06	96.58	96.8
micro Avg.	88.29	**88.77**	83.26	77.57	86.00	85.88	82.18	81.81
macro Avg.	88.291	**88.77**	83.257	77.573	85.99	85.882	82.18	81.809

Table 6: Accuracy obtained using SVM Light

Class	Avg. length	Avg. # non-zero features	Avg. percentage
acq	132.22	22.35	0.29%
corn	208.17	42.35	1.01%
crude	236.73	40.26	0.95%
earn	78.80	9.91	0.22%
grain	187.15	33.16	0.56%
interest	180.71	40.99	0.99%
money-fx	208.4	39.63	0.83%
ship	170.27	36	0.86%
trade	265.11	44.46	1.05%
wheat	184.11	37.42	0.86%

Table 7: Document statistics and feature counts information for DRN representation on the training set.

One of the classes with low F-score and Accuracy is *earn*, despite the fact that the class is the most balanced (40%+ positive instances in both the training and testing sets). Analysis of the texts in this class shows an average document length (in tokens) of 78.8 in the training set, out of which 67.91 words, and 62.12 in the testing set, with 50.08 words on average. This class contains most documents consisting of tables (62% of the *earn* instances in the training set, and 81% in the test set). Having a feature that indicates if a document consists of a table will lead to good predictions, but the feature is idiosyncratic, and it does not describe the semantic content of the document.

From such structured documents, the parser will not be able to produce informative word pairs. The average number of pairs for this class is a mere 9.91 (DRY) and 7.49 (DRN) for training, and 9.30 (DRY) and 7.08 (DRN) for the testing sets. These numbers correspond to less than 0.22% of the word-pair based feature vector length.

In order to obtain a more global view of the impact of text characteristics on learning performance, we look at two factors: average text lengths and average number of non-zero valued features in the representation of the documents. Table 7 shows the average document length and average vector sparseness (absolute and percentage) for each of the 10 classes we experimented with, the dependency-based DRN representation for the training set. We observe a high correlation (0.91) between document length and number of word-pair features extracted[7]. The plots connect points corresponding to the sparseness information and F-scores respectively. Although the individual points bare no relation to each other, we connect them to emphasize that an increase in vector coverage is closely mirrorred in many cases in an increase in F-score.

Figures 1 and 2 plot the average number of features and the F1-scores per class for the representation using dependency pair with no relation information (DRN), and for the unigram (U) representation respectively.

The classes are ordered on the x-axis in the increasing order of the percentage of positive instances in the training (and testing) sets. From inspecting the graphs we observe a strong correlation between the vector coverage for the DRN representation with the SVM F1-scores (0.96 correlation for the training set, 0.64 for testing).

[7] Feature counts were computed on the datasets generated after InfoGain filtering of features. When the full set of features is considered, the correlation remains.

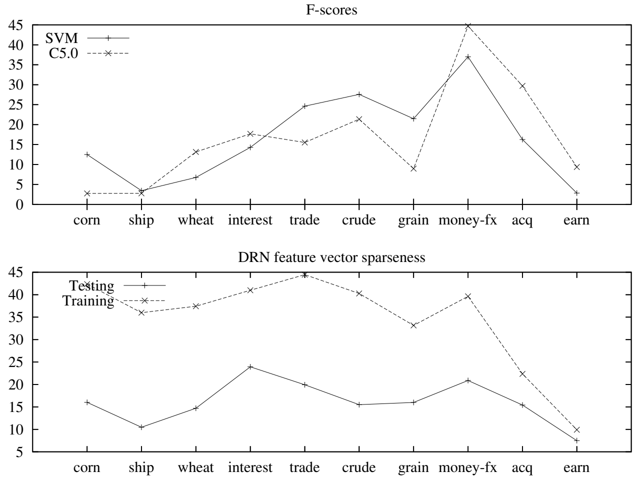

Figure 1: Vector sparseness and F1-scores for DRN representation

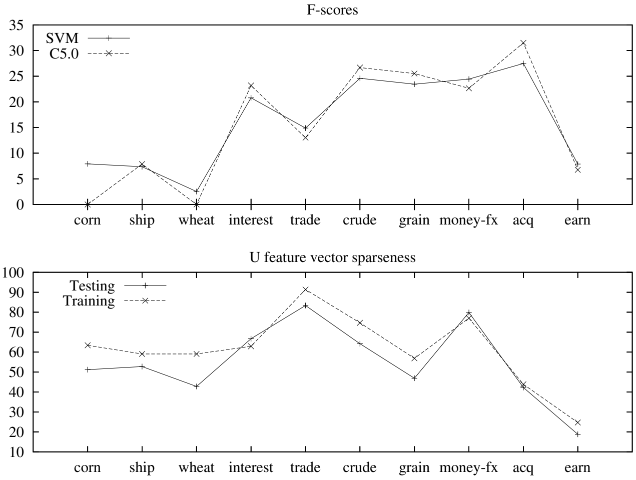

Figure 2: Vector sparseness and F1-scores for U representation

This seems to indicate that a sparser feature vector, caused also by text structure and limited length, leads to poorer F1-scores with SVM, than when the vector is less sparse.

6 Conclusions

Based on the results produced with two classification algorithms and the data analysis, we conclude that dependency pairs produce a representation that gives better results for text classification on the 10 most frequent classes of the Reuters-21578 corpus, in terms of precision and accuracy. We plan to investigate a range of experimental parameters such as feature selection methods (such as Mutual Information, shown to perform well on the Reuters-21578 collection [Dumais *et al.*, 1998; Joachim, 1998]) and performance assessment techniques – such as break even point, ROC graphs. We plan to experiment with other types of combined representation – using both syntactically motivated word pairs and unigrams – in which the feature selection is performed on the aggregated set of features, as opposed to combining the features for unigrams and syntactic pairs after feature selection.

Most of the documents in the Reuters-21578 collection are short, many of them consist solely of a table.

We expect to observe a more dramatic improvement in text classification using syntactically related word pairs for a different style of texts. We plan to verify this by applying the feature construction techniques described in the paper to other corpora, and assess their impact on classification performance.

We have seen that there is no clear winner from the two learning algorithms we have used. We plan to explore using other learning paradigms, and to test whether based on text characteristics – document length in tokens, number of features (unigrams and word pairs) – we can perform "meta-learning" – can we learn how to choose the type of learning algorithm that would work best.

References

M.F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Amita G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001.

H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of CIKM-98*, pages 148–155, 1998.

J. L. Fagan. *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*. PhD thesis, Ithaca, US, 1987.

C. J. Fillmore. The case for case. In E. Bach and R.T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Rinehart and Winston, 1968.

J. Furnkranz, T. M. Mitchell, and E. Rilof. A case study in using linguistic phrases for text categorization on the WWW. In *Proc. of the AAAI Workshop on Learning for Text Categorization*, pages 5–12, Madison, US, 1998.

J. Furnkranz. A study using n-gram features for text categorization. Technical Report Technical Report TR-98-30, Oesterreichisches Forschungsinstitut fur Artificial Intelligence, Wien, AT, 1998. http://www.ai.univie.ac.at/cgi-bin/tr-online?number+98-30.

J. Furnkranz. Exploiting structural information for text classification on the WWW. In *Proc. of IDA'99*, pages 487–497, Amsterdam, NL, 1999.

Sanda Harabagiu, Finley Lacatusu, and Andrew Hickl. Answering complex questions with random walk models. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 220–227, Seatle, WA, USA, 2006.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. Satisfying information needs with multi-document summaries. *Information Processing and Management*, 43(6):1619–1642, 2007.

S. Harabagiu. Incremental topic representations. In *In Proceedings of the 20th COLING Conference*, Geneva, Switzerland, 2004.

P. S. Jacobs. Joining statistics with nlp for text categorization. In *Proc. of the ANLP'92*, pages 178–185, Trento, Italy, 1992.

T. Joachim. Text categorization with support vector machines: learning with many redundant features. In *Proc. of ECML'98*, pages 137–142, Chemnitz, Germany, 1998.

D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In *Proc. of SIGIR-90*, pages 385–404, Bruxelles, BE, 1990.

D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proc. of SIGIR-92*, pages 37–50, New York, US, 1992.

D. D. Lewis. *Representation and Learning in Information Retrieval*. PhD thesis, 1992.

Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics, COLING-00*, pages 495–501, Saärbrücken, Germany, 2000.

D. Lin. Dependency-based evaluation of MiniPar. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, 1998.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *5TH RIAO Conference, Computer-Assisted Information Searching On the Internet*, pages 200–214, 1997.

D. Mladenic and M. Grobelnik. Word sequences as features in text learning. In *Proc. of ERK-98*, pages 145–148, Ljubljana, Slovenia, 1998.

C. Nedellec, M. Ould, F. Caropreso, P. Manine, and S. Matwin. Sentence categorization in genomics bibliography: a Nave Bayes approach. In *Informatique pour l'analyse du transcriptome*, Paris, 2003.

S. Scott and S. Matwin. Feature engineering for text classification. In *Proc. of ICML-99*, 1999.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

R.E. Shapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Computational Learning Theory*, pages 80–91, 1998.

S. M. Weiss, C. Apte, F. J. Darneau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. *IEEE Software Systems*, 14(4):63–69, 1999.