Automatic Domain Adaptation Outperforms Manual Domain Adaptation for Predicting Financial Outcomes

Marina Sedinkina[1] **Nikolas Breitkopf**[2] **Hinrich Schütze**[1]

[1]Center for Information & Language Processing, LMU Munich
[2]Institute for Finance & Banking, LMU Munich
sedinkina@cis.uni-muenchen.de

Abstract

In this paper, we automatically create sentiment dictionaries for predicting financial outcomes. We compare three approaches: (i) manual adaptation of the domain-general dictionary H4N, (ii) automatic adaptation of H4N and (iii) a combination consisting of first manual, then automatic adaptation. In our experiments, we demonstrate that the automatically adapted sentiment dictionary outperforms the previous state of the art in predicting the financial outcomes *excess return* and *volatility*. In particular, automatic adaptation performs better than manual adaptation. In our analysis, we find that annotation based on *an expert's a priori belief* about a word's meaning can be incorrect – annotation should be performed based on the word's *contexts in the target domain* instead.

1 Introduction

Since 1934, the U.S. Securities and Exchange Commission (SEC) mandates that public companies disclose information in form of public filings to ensure that adequate information is available to investors. One such filing is the 10-K, the company's annual report. It contains financial statements and information about business strategy, risk factors and legal issues. For this reason, 10-Ks are an important source of information in the field of finance and accounting.

A common method employed by finance and accounting researchers is to evaluate the "tone" of a text based on the Harvard Psychosociological Dictionary, specifically, on the Harvard-IV-4 TagNeg (H4N) word list.[1] However, as its name suggests, this dictionary is from a domain that is different from finance, so many words (e.g., "liability", "tax") that are labeled as negative in H4N are in fact not negative in finance.

In a pioneering study, Loughran and Mcdonald (2011) manually reclassified the words in H4N for the financial domain. They applied the resulting dictionaries[2] to 10-Ks and predicted financial variables such as excess return and volatility. We will refer to the sentiment dictionaries created by Loughran and Mcdonald (2011) as L&M.

In this work, we also create sentiment dictionaries for the finance domain, but we adapt them from the domain-general H4N dictionary *automatically*. We first learn word embeddings from a corpus of 10-Ks and then reclassify them – using SVMs trained on H4N labels – as negative vs. non-negative. We refer to the resulting domain-adapted dictionary as $H4N_{RE}$.

In our experiments, we demonstrate that the automatically adapted financial sentiment dictionary $H4N_{RE}$ performs better at predicting excess return and volatility than dictionaries of Loughran and Mcdonald (2011) and Theil et al. (2018).

We make the following contributions. **(i)** We demonstrate that automatic domain adaptation performs better at predicting financial outcomes than previous work based on manual domain adaptation. **(ii)** We perform an analysis of the differences between the classifications of L&M and those of our sentiment dictionary $H4N_{RE}$ that sheds light on the superior performance of $H4N_{RE}$. For example, $H4N_{RE}$ is much smaller than L&M, consisting mostly of frequent words, suggesting $H4N_{RE}$ is more robust and less prone to overfitting. **(iii)** In a further detailed analysis, we investigate words classified by L&M as *negative*, *litigious* and *uncertain* that our embedding classifier classifies otherwise; and common (i.e., non-negative) words from H4N that L&M did not include in the categories *negative*, *litigious* and *uncertain*, but that our embedding classifier classifies as belonging to these classes. Our analysis suggests that manual

[1]http://www.wjh.harvard.edu/~inquirer

[2]https://sraf.nd.edu/textual-analysis/resources

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 346–359
Florence, Italy, July 28 - August 2, 2019.

adaptation of dictionaries is error-prone if annotators are not given access to corpus contexts.

Our paper primarily addresses a finance application. In empirical finance, a correct sentiment classification decision is not sufficient – the decision must also be *interpretable* and *statistically sound*. That is why we use ordinary least squares (OLS) – an established method in empirical finance – and sentiment dictionaries. Models based on sentiment dictionaries are transparent and interpretable: by looking at the dictionary words occurring in a document we can trace the classification decision back to the original data and, e.g., understand the cause of a classification error. OLS is a well-understood statistical method that allows the analysis of significance, effect size and dependence between predictor variables, inter alia.

While we focus on finance here, three important lessons of our work also apply to many other domains. (1) An increasing number of applications require interpretable analysis; e.g., the European Union mandates that systems used for sensitive applications provide explanations of decisions. Decisions based on a solid statistical foundation are more likely to be trusted than those by black boxes. (2) Many NLP applications are domain-specific and require domain-specific resources including lexicons. Should such lexicons be built manually from scratch or adapted from generic lexicons? We provide evidence that automatic adaptation works better. (3) Words often have specific meanings in a domain and this increases the risk that a word is misjudged if only the generic meaning is present to the annotator. This seems to be the primary reason for the problems of manual lexicons in our experiments. Thus, if manual lexicon creation is the only option, then it is important to present words in context, not in isolation, so that the domain-specific sense can be recognized.

2 Related Work

In **empirical finance**, researchers have exploited various text resources, e.g., news (Kazemian et al., 2016), microblogs (Cortis et al., 2017), twitter (Zamani and Schwartz, 2017) and company disclosures (Nopp and Hanbury, 2015; Kogan et al., 2009). Deep learning has been used for learning document representations (Ding et al., 2015; Akhtar et al., 2017). However, the methodology of empirical finance requires interpretable results. Thus, a common approach is to define features for statistical models like Ordinary Least Squares (Lee et al., 2014; Rekabsaz et al., 2017). Frequently, lexicons like H4N TagNeg[3] (Tetlock et al., 2007) are used. It includes a total of 85,221 words, 4188 of which are labeled negative. The remaining words are labeled "common", i.e., non-negative. Loughran and Mcdonald (2011) argue that many words from H4N have a specialized meaning when appearing in an annual report. For instance, domain-general negative words such as "tax", "cost", "liability" and "depreciation" – which predominate in 10-Ks – do not typically have negative sentiment in 10-Ks. So Loughran and Mcdonald (2011) constructed subjective financial dictionaries manually, by examining all words that appear in at least 5% of 10-Ks and classifying them based on their assessment of most likely usage. More recently, other finance-specific lexicons were created (Wang et al., 2013). Building on L&M, Tsai and Wang (2014) and Theil et al. (2018) show that the L&M dictionaries can be further improved by adding most similar neighbors to words manually labeled by L&M.

Seed-based methods generalize a set of seeds based on corpus (e.g., distributional) evidence. Models use syntactic patterns (Hatzivassiloglou and McKeown, 1997; Widdows and Dorow, 2002), cooccurrence (Turney, 2002; Igo and Riloff, 2009) or label propagation on lexical graphs derived from cooccurrence (Velikovich et al., 2010; Huang et al., 2014).

Supervised methods start with a larger training set, not just a few seeds (Mohammad et al., 2013). Distributed word representations (Tang et al., 2014; Amir et al., 2015; Vo and Zhang, 2016; Rothe et al., 2016) are beneficial in this approach. For instance, Tang et al. (2014) incorporate in word embeddings a document-level sentiment signal. Wang and Xia (2017) also integrate document and word levels. Hamilton et al. (2016) learn domain-specific word embeddings and derive word lists specific to domains, including the finance domain.

Dictionary-based approaches (Takamura et al., 2005; Baccianella et al., 2010; Vicente et al., 2014) use hand-curated lexical resources – often WordNet (Fellbaum, 1998) – for constructing lexicons. Hamilton et al. (2016) argue that dictionary-based approaches generate better re-

[3] http://www.wjh.harvard.edu/~inquirer/

sults due to the quality of hand-curated resources. We compare two ways of using a hand-curated resource in this work – a general-domain resource that is automatically adapted to the specific domain vs. a resource that is manually created for the specific domain – and show that automatic domain adaptation performs better.

Apart from domain adaptation work on dictionaries, many other approaches to **generic domain adaptation** have been proposed. Most of this work adopts the classical domain adaptation scenario: there is a large labeled training set available in the source domain and an amount of labeled target data that is insufficient for training a high-performing model on its own (Blitzer et al., 2006; Chelba and Acero, 2006; Daumé III, 2009; Pan et al., 2010; Glorot et al., 2011; Chen et al., 2012). More recently, the idea of domain-adversarial training was introduced for the same scenario (Ganin et al., 2016). In contrast to this work, we do not transfer any parameters or model structures from source to target. Instead, we use labels from the source domain and train new models from scratch based on these labels: first embedding vectors, then a classifier that is trained on source domain labels and finally a regression model that is trained on the classification decisions of the classifier. This approach is feasible in our problem setting because the divergence between source and target sentiment labels is relatively minor, so that training target embeddings with source labels gives good results.

The motivation for this different setup is that our work primarily addresses a finance application where explainability is of high importance. For this reason, we use a model based on sentiment dictionaries that allows us to provide explanations of the model's decisions and predictions.

3 Methodology

3.1 Empirical finance methodology

In this paper, we adopt Ordinary Least Squares (OLS), a common research method in empirical finance: a dependent variable of interest (e.g., excess return, volatility) is predicted based on a linear combination of a set of explanatory variables.

The main focus of this paper is to investigate text-based explanatory variables: we would like to know to what extent a text variable such as occurrence of negative words in a 10-K can predict a financial variable like volatility. Identifying the economic drivers of such a financial outcome is of central interest in the field of finance. Some of these determinants may be correlated with sentiment. To understand the role of sentiment in explaining financial variables we therefore need to isolate the *complementary information* of our text variables. This is achieved by including in our regressions – as control variables – a standard set of financial explanatory variables such as firm size and book-to-market ratio. These control variables are added as additional explanatory variables in the regression specification besides the textual sentiment variables. This experimental setup allows us to assess the added benefit of text-based variables in a realistic empirical finance scenario.

The approach is motivated by previous studies in the finance literature (e.g., Loughran and Mcdonald (2011)), which show that characteristics of financial firms can explain variation in excess returns and volatility. By including these control variables in the regression we are able to determine whether sentiment factors have incremental explanatory power beyond the already established financial factors. Since the inclusion of these control variables is not primarily driven by the assumption that firms with different characteristics use different language, our approach differs from other NLP studies, such as Hovy (2015), who accounts for non-textual characteristics by training group-specific embeddings.

Each text variable we use is based on a dictionary. Its value for a 10-K is the proportion of tokens in the 10-K that are members of the dictionary. For example, if the 10-K is 5000 tokens long and 50 of those tokens are contained in the L&M uncertainty dictionary, then the value of the L&M uncertainty text variable for this 10-K is 0.01.

In the type of analysis of stock market data we conduct, there are two general forms of dependence in the residuals of a regression, which arise from the panel structure of our data set where a single firm is repeatedly observed over time and multiple firms are observed at the same point in time. *Firm effect:* Time-series dependence assumes that the residuals of *a given firm* are correlated *across years*. *Time effect:* Cross-sectional dependence assumes that the residuals of *a given year* are correlated *across different firms*. These properties violate the i.i.d. assumption of residuals in standard OLS. We therefore model data with both firm and time effects and run a *two-*

way robust cluster regression, i.e., an OLS regression with standard errors that are clustered on two dimensions (Gelbach et al., 2009), the dimensions of firm and time.[4] We apply this regression-based methodology to test the explanatory power of financial dictionaries with regard to two dependent variables: excess return and volatility. This approach allows us to compare the explanatory power of different sentiment dictionaries and in the process test the hypothesis that negative sentiment is associated with subsequently lower stock returns and higher volatility. We now introduce the regression specifications for these tests.

3.1.1 Excess return

The dependent variable excess return is defined as the firm's buy-and-hold stock return minus the value-weighted buy-and-hold market index return during the 4-day event window starting on the 10-K filing date, computed from prices by the Center for Research in Security Prices (CRSP)[5] (both expressed as a percentage). In addition to the independent text variables (see §4 for details), we include the following financial control variables. (i) Firm size: the log of the book value of total assets. (ii) Alpha of a Fama-French regression (Fama and French, 1993) calculated from days [-252 -6];[6] this represents the "abnormal" return of the asset, i.e., the part of the return not due to common risk factors like market and firm size. (iii) Book-to-market ratio: the log of the book value of equity divided by the market value of equity. (iv) Share turnover: the volume of shares traded in days [-252 -6] divided by shares outstanding on the filing date. (v) Earnings surprise, computed by IBES from Thomson Reuters;[7] this variable captures whether the reported financial performance was better or worse than expected by financial analysts.[8]

[4] Loughran and Mcdonald (2011) use the method of Fama and MacBeth (1973) instead. This method assumes that the yearly estimates of the coefficient are independent of each other. However, this is not true when there is a firm effect.

[5] http://www.crsp.com

[6] [-252 -6] is the notation for the 252 days prior to the filing date with the last 5 days prior to the filing date excluded.

[7] http://www.thomsonreuters.com

[8] Our setup largely mirrors, but is not identical to the one used by Loughran and Mcdonald (2011) because not all data they used are publicly available and because we use a larger time window (1994-2013) compared to theirs (1994-2008).

dictionary	size
neg_{lm}	2355
unc_{lm}	297
lit_{lm}	903
neg_{ADD}	2340
unc_{ADD}	240
lit_{ADD}	984
neg_{RE}	1205
unc_{RE}	96
lit_{RE}	208
H4N_{ORG}	4188
H4N_{RE}	338

Table 1: Number of words per dictionary

3.1.2 Volatility

The dependent variable volatility is defined as the post-filing root-mean-square error (RMSE) of a Fama-French regression calculated from days [6 252]. The RMSE captures the idiosyncratic component of the total volatility of the firm, since it picks up the stock price variation that cannot be explained by fluctuations of the common risk factors of the Fama-French model. The RMSE is therefore a measure of the financial uncertainty of the firm. In addition to the independent text variables (see §4 for details), we include the following financial control variables. (i) Pre-filing RMSE and (ii) pre-filing alpha of a Fama-French regression calculated from days [-252 -6]; these characterize the financial uncertainty and abnormal return of the firm in the past (see §3.1.1 for alpha and first sentence of this section for RMSE). (iii) Filing abnormal return; the value of the buy-and-hold return in trading days [0 3] minus the buy-and-hold return of the market index. (iv) Firm size and (v) book-to-market ratio (the same as in §3.1.1). (vi) Calendar year dummies and Fama-French 48-industry dummies to allow for time and industry fixed effects.[9]

3.2 NLP methodology

There are two main questions we want to answer:

Q1. Is a manually domain-adapted or an automatically domain-adapted dictionary a more effective predictor of financial outcomes?

Q2. L&M adapted H4N for the financial domain and showed that this manually adapted dictionary is more effective than H4N for prediction. Can we further improve L&M's manual adaptation

[9] We do not include in the regression a Nasdaq dummy variable indicating whether the firm is traded on Nasdaq. Since Nasdaq mainly lists tech companies, the Nasdaq effect is already captured by industry dummies.

by automatic domain adaptation?

The general methodology we employ for domain adaptation is based on word embeddings. We train CBOW word2vec (Mikolov et al., 2013) word embeddings on a corpus of 10-Ks for all words of H4N that occur in the corpus – see §4 for details. We consider two adaptations: ADD and RE. ADD is only used to answer question Q2.

ADD. For adapting the L&M dictionary, we train an SVM on an L&M dictionary in which words are labeled +1 if they are marked for the category by L&M and labeled -1 otherwise (where the category is negative, uncertain or litigious). Each word is represented as its embedding. We then run the SVM on all H4N words that are not contained in the L&M dictionary. We also ignore H4N words that we do not have embeddings for because their frequency is below the word2vec frequency threshold. Thus, we obtain an ADD dictionary which is not a superset of the L&M lexicon because it includes only new additional words that are not part of the original dictionary.

SVM scores are converted into probabilities via logistic regression. We define a confidence threshold θ – we only want to include words in the ADD dictionary that are reliable indicators of the category of interest. A word is added to the dictionary if its converted SVM score is greater than θ.

RE. We train SVMs as for ADD, but this time in a five-fold cross validation setup. Again, SVM scores are converted into probabilities via logistic regression. A word w becomes a member of the adapted dictionary if its converted SVM score of the SVM that was not trained on the fold that contains w is greater than θ.

To answer our first question Q1: "Is automatic or manual adaptation better?", we apply adaptation method RE to H4N and compare the results to the L&M dictionaries.

To answer our second question Q2: "Can manual adaptation be further improved by automatic adaptation?", we apply adaptation methods RE and ADD to the three dictionaries compiled by L&M and compare results for original and adapted L&M dictionaries: (i) negative (abbreviated as "neg"), (ii) uncertain (abbreviated as "unc"), (iii) litigious (abbreviated as "lit"). Our goals here are to improve the in-domain L&M dictionaries by relabeling them using adaptation method RE and to find new additional words using adaptation method ADD.

Table 1 gives dictionary sizes.

4 Experiments and results

We downloaded 206,790 10-Ks for years 1994 to 2013 from the SEC's database EDGAR.[10] Table of contents, page numbers, links and numeric tables are removed in preprocessing and only the main body of the text is retained. Documents are split into sections. Sections that are not useful for textual analysis (e.g., boilerplate) are deleted.

To construct the final sample, we apply the filters defined by L&M (Loughran and Mcdonald, 2011): we require a match with CRSP's permanent identifier PERMNO, the stock to be common equity, a stock pre-filing price of greater than $3, a positive book-to-market, as well as CRSP's market capitalization and stock return data available at least 60 trading days before and after the filing date. We only keep firms traded on Nasdaq, NYSE or AMEX and whose filings contain at least 2000 words. This procedure results in a corpus of 60,432 10-Ks. We tokenize (using NLTK) and lowercase this corpus and remove punctuation.

We use word2vec CBOW with hierarchical softmax to learn word embeddings from the corpus. We set the size of word vectors to 400 and run one training iteration; otherwise we use word2vec's default hyperparameters. SVMs are trained on word embeddings as described in §3.2. We set the threshold θ to 0.8, so only words with converted SVM scores greater than 0.8 will be added to dictionaries.[11]

As described in §3, we compare manually adapted and automatically adapted dictionaries (Q1) and investigate whether automatic adaptation of manually adapted dictionaries further improves performance (Q2). Our experimental setup is Ordinary Least Squares (OLS), more specifically, a two-way robust cluster regression for the time and firm effects. The dependent financial variable is excess return or volatility. We include several independent financial variables in the regression as well as one or more text variables. The value of the text variable for a category is the proportion of tokens from the category that occur in a 10-K.

To assess the utility of a text variable for predicting a financial outcome, we look at significance and the standardized regression coefficient

[10] https://www.sec.gov/edgar.shtml

[11] We choose this threshold because the proportion of negative, litigious and uncertain words in 10-Ks for 0.8 is roughly the same as when using L&M dictionaries.

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
lit_{lm}	-0.0291	-0.026	-0.83	1.00
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
H4N_{RE}	-0.764***	*-0.229*	**-3.04**	1.05

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 2: Excess return regression results for L&M dictionaries and reclassified H4N dictionary. **For all tables in this paper, significant t values are bolded and best standard coefficients per category are in italics.**

var	coeff	std coeff	t	R^2
H4N_{RE}	-0.88**	*-0.264*	**-2.19**	1.05
neg_{lm}	0.062	0.024	0.48	
H4N_{RE}	-0.757***	-0.227	**-2.90**	1.05
lit_{lm}	-0.351	*-0.315*	-0.013	
H4N_{RE}	-0.746***	*-0.223*	**-2.89**	1.05
unc_{lm}	-0.45	-0.135	-0.45	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 3: Excess return regression results for multiple text variables. This table shows results for three regressions that combine H4N_{RE} with each of the three L&M dictionaries.

(the product of regression coefficient and standard deviation). If a result is significant, then it is unlikely that the result is due to chance. The standardized coefficient measures the effect size, normalized for different value ranges of variables. It can be interpreted as the expected change in the dependent variable if the independent variable increases by one standard deviation. The standardized coefficient allows a fair comparison between a text variable that, on average, has high values (many tokens per document) with one that, on average, has low values (few tokens per document).

4.1 Excess Return

Table 2 gives regression results for excess return, comparing H4N_{RE} (our automatic adaptation of the general Harvard dictionary) with the three manually adapted L&M dictionaries. As expected the coefficients are negatively signed – 10-Ks containing a high percentage of pessimistic words are associated with negative excess returns.

L&M designed the dictionary neg_{lm} specifically for measuring negative information in a 10-K that may have a negative effect on outcomes like excess return. So it is not surprising that neg_{lm} is the best performing dictionary of the three L&M dictionaries: it has the highest standard coefficient (-0.080) and the highest significance (-2.56). unc_{lm} performs slightly worse, but is also significant.

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
neg_{spec}	0.0102	0.0132	0.27	1.00
neg_{RE}	-0.37***	*-0.111*	**-2.96**	1.03
neg_{ADD}	-0.033	-0.0231	-1.03	1.00
$\text{neg}_{\text{RE+ADD}}$	-0.08**	-0.072	**-2.19**	1.03
lit_{lm}	-0.0291	-0.026	-0.83	1.00
lit_{RE}	-0.056	*-0.028*	-0.55	1.00
lit_{ADD}	-0.0195	-0.0156	-0.70	1.00
$\text{lit}_{\text{RE+ADD}}$	-0.0163	-0.0211	-0.69	1.00
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
unc_{RE}	-0.377***	*-0.075*	**-2.77**	1.02
unc_{ADD}	0.0217	0.0065	0.21	1.00
$\text{unc}_{\text{RE+ADD}}$	-0.0315	-0.0157	-0.45	1.00

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 4: Excess return regression results for L&M, **RE** and **ADD** dictionaries

However, when comparing the three L&M dictionaries with H4N_{RE}, the automatically adapted Harvard dictionary, we see that H4N_{RE} performs clearly better: it is highly significant and its standard coefficient is larger by a factor of more than 2 compared to neg_{lm}. This evidence suggests that the automatically created H4N_{RE} dictionary has a higher explanatory power for excess returns than the manually created L&M dictionaries. This provides an initial answer to question Q1: in this case, automatic adaptation beats manual adaptation.

Table 3 shows *manual plus automatic* experiments with *multiple* text variables in one regression, in particular, the combination of H4N_{RE} with each of the L&M dictionaries. We see that the explanatory power of L&M variables is lost after we additionally include H4N_{RE} in a regression: all three L&M variables are not significant. In contrast, H4N_{RE} continues to be significant in all experiments, with large standard coefficients. More manual plus automatic experiments can be found in the appendix. These experiments further confirm that automatic is better than manual adaptation.

Table 4 shows results for automatically adapting the L&M dictionaries.[12] The subscript "RE+ADD" refers to a dictionary that merges RE and ADD; e.g., $\text{neg}_{\text{RE+ADD}}$ is the union of neg_{RE} and neg_{ADD}.

We see that for each category (neg, lit and unc), the automatically adapted dictionary performs better than the original manually adapted dictionary; e.g., the standard coefficient of neg_{RE} is -0.111,

[12] Experiments with multiple text variables in one regression (manual plus automatic experiments) are presented in the appendix.

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
unc_{lm}	0.119*	0.0356	**2.25**	60.0
H4N_{RE}	0.577***	*0.173*	**4.40**	60.3

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 5: Volatility regression results for L&M dictionaries and reclassified H4N dictionary

var	coeff	std coeff	t	R^2
H4N_{RE}	0.748***	*0.224*	**4.44**	1.11
neg_{lm}	-0.096*	-0.038	-2.55	
H4N_{RE}	0.642***	*0.192*	**4.28**	1.11
lit_{lm}	-0.041*	-0.037	-2.54	
H4N_{RE}	0.695***	0.208	**4.54**	1.11
unc_{lm}	-0.931**	*-0.279*	-2.73	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 6: Volatility regression results for multiple text variables

clearly better than that of neg_{lm} (-0.080). Results are significant for neg_{RE} (-2.96) and unc_{RE} (-2.77). We also evaluate neg_{spec}, the negative word list of Hamilton et al. (2016). neg_{spec} does not perform well: it is not significant.

These results provide a partial answer to question Q2: for excess return, automatic adaptation of L&M's manually adapted dictionaries further improves their performance.

4.2 Volatility

Table 5 compares H4N_{RE} and L&M regression results for volatility. Except for litigious, the coefficients are positive, so the greater the number of pessimistic words, the greater the volatility.

Results for neg_{lm}, unc_{lm} and H4N_{RE} are statistically significant. The best L&M dictionary is again neg_{lm} with standard coefficient 0.0472 and $t = 3.30$. However, H4N_{RE} has the highest explanatory value for volatility. Its standard coefficient (0.173) is more than three times as large as that of neg_{lm}.

The higher effect size demonstrates that H4N_{RE} better explains volatility than the L&M dictionaries. Again, this indicates – answering question Q1 – that automatic outperforms manual adaptation. Table 6 confirms this. We see that for manual plus automatic experiments each combination of H4N_{RE} with one of the L&M dictionaries provides significant results for H4N_{RE}. In contrast, L&M dictionaries become negatively signed meaning that more uncertain words decrease volatility, suggesting that they are not indicative of the true relationship between volatility and negative tone in 10-Ks in this regression setup. Our results of additional manual plus automatic experiments support this observation as well. See the appendix for an illustration.

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
neg_{spec}	-0.038	-0.0494	-2.73	60.1
neg_{RE}	0.219***	*0.0657*	**3.57**	60.1
neg_{ADD}	0.032***	0.0224	**4.06**	60.0
$\text{neg}_{\text{RE+ADD}}$	0.038***	0.0342	**4.32**	60.1
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
lit_{RE}	0.0080	0.0040	0.20	60.0
lit_{ADD}	0.028	*0.0224*	1.07	60.0
$\text{lit}_{\text{RE+ADD}}$	0.015	0.0195	0.81	60.0
unc_{lm}	0.119*	*0.0356*	**2.25**	60.0
unc_{spec}	-0.043	-0.0344	-1.56	60.0
unc_{RE}	0.167*	0.0334	**2.30**	60.0
unc_{ADD}	-0.013	-0.0039	-0.17	60.0
$\text{unc}_{\text{RE+ADD}}$	0.035	0.0175	0.68	60.0

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 7: Volatility regression results for L&M, **RE** and **ADD** dictionaries

Table 7 gives results for automatically adapting the L&M dictionaries.[13] For neg, the standard coefficient of neg_{RE} is 0.0657, better by about 40% than neg_{lm}'s standard coefficient of 0.0472. neg_{spec} does not provide significant results and has the negative sign, i.e., an increase of negative words decreases volatility. The lit dictionaries are not significant (neither L&M nor adapted dictionaries). For unc, unc_{RE} performs worse than unc_{lm}, but only slightly by 0.0344 vs. 0.0356 for the standard coefficients. The overall best result is neg_{RE} (standard coefficient 0.0657). Even though L&M designed the unc_{lm} dictionary specifically for volatility, our results indicate that neg dictionaries perform better than unc dictionaries, both for L&M dictionaries (neg_{lm}) and their automatic adaptations (e.g., neg_{RE}).

Table 7 also evaluates unc_{spec}, the uncertainty dictionary of Theil et al. (2018). unc_{spec} does not perform well: it is not significant and the coefficient has the "wrong" sign.[14]

The main finding supported by Table 7 is that

[13] Experiments with multiple text variables in one regression (manual plus automatic experiments) are presented in the appendix.

[14] Theil et al. (2018) define volatility for the time period [6 28] whereas our definition is [6 252], based on (Loughran and Mcdonald, 2011). Larger time windows allow more reliable estimates and account for the fact that information disclosures can influence volatility for long periods (Belo et al., 2016).

ADD_{neg}	missing, diminishment, disabling, overuse
ADD_{unc}	reevaluate, swings, expectation, estimate
ADD_{lit}	lender, assignors, trustee, insurers
RE_{neg}	confusion, unlawful, convicted, breach
RE_{unc}	variability, fluctuation, variations, variation
RE_{lit}	courts, crossclaim, conciliation, abeyance
$H4N_{RE}$	compromise, issues, problems, impair, hurt

Table 8: Word classification examples from automatically adapted dictionaries

the best automatic adaptation of an L&M dictionary gives rise to more explanatory power than the best L&M dictionary, i.e., neg_{RE} performs better than neg_{lm}. This again confirms our answer to Q2: we can further improve manual adaptation by automatic domain adaptation.

5 Analysis and discussion

5.1 Qualitative Analysis

Our dictionaries outperform L&M. In this section, we perform a qualitative analysis to determine the reasons for this discrepancy in performance.

Table 8 shows words from automatically adapted dictionaries. Recall that the **ADD** method adds words that L&M classified as nonrelevant for a category. So words like "missing" (neg), "reevaluate" (unc) and "assignors" (lit) were classified as relevant terms and seem to connote negativity, uncertainty and litigiousness, respectively, in financial contexts.

In L&M's classification scheme, a word can be part of several different categories. For instance, L&M label "unlawful", "convicted" and "breach" both as litigious and as negative. When applying our RE method, these words were only classified as negative, not as litigious. Similarly, L&M label "confusion" as negative and uncertain, but automatic RE adaptation labels it only negative. This indicates that there is strong distributional evidence in the corpus for the category negativity, but weaker distributional evidence for litigious and uncertain. For our application, only "negative" litigious/uncertain words are of interest – "acquittal" (positive litigious) and "suspense" (positive uncertain) are examples of positive words that may not help in predicting financial variables. This could explain why the negative category fares better in our adaptation than the other two.

An interesting case study for RE is "abeyance". L&M classify it as uncertain, automatic adaptation as litigious. Even though "abeyance" has a domain-general uncertain sense ("something that is waiting to be acted upon"), it is mostly used in legal contexts in 10-Ks: "held in abeyance", "appeal in abeyance". The nearest neighbors of "abeyance" in embedding space are also litigious words: "stayed", "hearings", "mediation".

$H4N_{RE}$ contains 74 words that are "common" in H4N. Examples include "compromise", "serious" and "god". The nearest neighbors of "compromise" in the 10-K embedding space are the negative terms "misappropriate", "breaches", "jeopardize". In a general-domain embedding space,[15] the nearest neighbors of "compromise" include "negotiated settlement", "accord" and "modus vivendi". This example suggests that "compromise" is used in 10-Ks in negative contexts and in the general domain in positive contexts. This also illustrates the importance of domain-specific word embeddings that capture domain-specific information.

Another interesting example is the word "god"; it is frequently used in 10-Ks in the phrase "act of God". Its nearest neighbors in the 10-K embedding space are "terrorism" and "war". This example clearly demonstrates that annotators are likely to make mistakes when they annotate words for sentiment without seeing their contexts. Most annotators would annotate "god" as positive, but when presented with the typical context in 10-Ks ("act of God"), they would be able to correctly classify it.

We conclude that manual annotation of words without context based on the prior belief an annotator has about word meanings is error-prone. Our automatic adaptation is performed based on the word's contexts in the target domain and therefore not susceptible to this type of error.

5.2 Quantitative Analysis

Table 9 presents a quantitative analysis of the distribution of words over dictionaries. For a row dictionary d_r and a column dictionary d_c, a cell gives $|d_r \cap d_c|/|d_r|$ as a percentage. (Diagonal entries are all equal to 100% and are omitted for space reasons.) For example, 49% of the words in neg_{lm} are also members of neg_{RE} (row "neg_{lm}", column "neg_{RE}"). This analysis allows us to obtain insights into the relationship between different dictionaries and into the relationship between

[15] https://code.google.com/archive/p/word2vec/

	neg_{lm}	lit_{lm}	unc_{lm}	neg_{ADD}	lit_{ADD}	unc_{ADD}	neg_{RE}	lit_{RE}	unc_{RE}	$H4N_{neg}$	$H4N_{cmn}$	$H4N_{RE}$
neg_{lm}		7	2	0	0	0	49	2	0	48	52	12
lit_{lm}	17		0	0	0	0	6	20	0	7	93	1
unc_{lm}	14	0		0	0	0	18	2	30	16	84	2
neg_{ADD}	0	0	0		0	0	0	0	0	18	82	2
lit_{ADD}	0	0	0	0		0	0	0	0	1	99	0
unc_{ADD}	0	0	0	0	0		0	0	0	3	97	0
neg_{RE}	95	5	4	0	0	0		0	1	52	48	21
lit_{RE}	18	86	2	0	0	0	0		0	7	93	0
unc_{RE}	11	2	92	0	0	0	10	0		13	87	3
$H4N_{neg}$	27	2	1	10	0	0	15	0	0		0	6
$H4N_{cmn}$	2	1	0	2	1	0	1	0	0	0		0
$H4N_{RE}$	79	2	2	17	0	0	74	0	1	78	22	

Table 9: Quantitative analysis of dictionaries. For a row dictionary d_r and a column dictionary d_c, a cell gives $|d_r \cap d_c|/|d_r|$ as a percentage. Diagonal entries (all equal to 100%) omitted for space reasons. cmn = common

the categories negative, litigious and uncertain.

Looking at rows neg_{lm}, lit_{lm} and unc_{lm} first, we see how L&M constructed their dictionaries. neg_{lm} words come from $H4N_{neg}$ and $H4N_{cmn}$ in about equal proportions; i.e., many words that are "common" in ordinary usage were classified as negative by L&M for financial text. Relatively few lit_{lm} and unc_{lm} words are taken from $H4N_{neg}$, most are from $H4N_{cmn}$. Only 12% of neg_{lm} words were automatically classified as negative in domain adaptation and assigned to $H4N_{RE}$. This is a surprisingly low number. Given that $H4N_{RE}$ performs better than neg_{lm} in our experiments, this statistic casts serious doubt on the ability of human annotators to correctly classify words for the type of sentiment analysis that is performed in empirical finance if the actual corpus contexts of the words are not considered. We see two types of failures in the human annotation. First, as discussed in §5.1, words like "god" are misclassified because the prevalent context in 10-Ks ("act of God") is not obvious to the annotator. Second, the utility of a word is not only a function of its sentiment, but also of the strength of this sentiment. Many words in neg_{lm} that were deemed neutral in automatic adaptation are probably words that may be slightly negative, but that do not contribute to explaining financial variables like excess return. The strength of sentiment of a word is difficult to judge by human annotators. Looking at the row $H4N_{RE}$, we see that most of its words are taken from neg_{lm} (79%) and a few from lit_{lm} and unc_{lm} (2% each). We can interpret this statistic as indicating that L&M had high recall (they found most of the reliable indicators), but low precision (see the previous paragraph: only 12% of their negative words survive in $H4N_{RE}$). The distribution of $H4N_{RE}$ words over $H4N_{neg}$ and $H4N_{cmn}$ is 78:22. This confirms the need for domain adaptation: many general-domain common words are negative in the financial domain.

We finally look at how dictionaries for negative, litigious and uncertain overlap, separately for the L&M, ADD and RE dictionaries. lit_{lm} and unc_{lm} have considerable overlap with neg_{lm} (17% and 14%), but they do not overlap with each other. The three ADD dictionaries – neg_{ADD}, lit_{ADD} and unc_{ADD} – do not overlap at all. As for RE, 10% of the words of unc_{RE} are also in neg_{RE}, otherwise there is no overlap between RE dictionaries. Comparing the original L&M dictionaries and the automatically adapted ADD and RE dictionaries, we see that the three categories – negative, litigious and uncertain – are more clearly distinguished after adaptation. L&M dictionaries overlap more, ADD and RE dictionaries overlap less.

6 Conclusion

In this paper, we automatically created sentiment dictionaries for predicting financial outcomes. In our experiments, we demonstrated that the automatically adapted sentiment dictionary $H4N_{RE}$ outperforms the previous state of the art in predicting the financial outcomes excess return and volatility. In particular, automatic adaptation performs better than manual adaptation. Our quantitative and qualitative study provided insight into the semantics of the dictionaries. We found that annotation based on *an expert's a priori belief* about a word's meaning can be incorrect – annotation should be performed based on the word's *contexts in the target domain* instead. In the future, we plan to investigate whether there are changes over time that significantly impact the linguistic characteristics of the data, in the simplest case changes in the meaning of a word. Another interesting topic for future research is the comparison of domain adaptation based on our domain-specific word embeddings vs. based on word embeddings trained on much larger corpora.

Acknowledgments

We are grateful for the support of the European Research Council for this work (ERC #740516).

References

Md Shad Akhtar, Abhishek Kumar, Deepanway Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. 2017. A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 540–546. Association for Computational Linguistics.

Silvio Amir, Wang Ling, Ramón Fernández Astudillo, Bruno Martins, Mário J. Silva, and Isabel Trancoso. 2015. INESC-ID: A regression model for large scale twitter sentiment lexicon induction. In *SemEval@NAACL-HLT*, pages 613–618. The Association for Computer Linguistics.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. European Language Resources Association.

Frederico Belo, Jun Li, Xiaoji Lin, and Xiaofei Zhao. 2016. Complexity and information content of financial disclosures: Evidence from evolution of uncertainty following 10-k filings. *SSRN*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

Ciprian Chelba and Alex Acero. 2006. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535. Association for Computational Linguistics.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2327–2333. AAAI Press.

Eugene F. Fama and Kenneth R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3 – 56.

Eugene F Fama and James D MacBeth. 1973. Risk, return, and equilibrium: Empirical tests. *Journal ter%20Dictionaryof political economy*, 81(3):607–636.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Jonah B Gelbach, Doug Miller, et al. 2009. Robust inference with multi-way clustering. Technical report, National Bureau of Economic Research.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. *CoRR*, abs/1606.02820.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, EACL '97, pages 174–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 752–762.

Sheng Huang, Zhendong Niu, and Chongyang Shi. 2014. Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. *Knowl.-Based Syst.*, 56:191–200.

Sean P. Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, UMSLLS '09, pages 18–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Siavash Kazemian, Shunan Zhao, and Gerald Penn. 2016. Evaluating sentiment analysis in the context of securities trading. In *ACL (1)*. The Association for Computer Linguistics.

Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. Predicting risk from financial reports with regression. In

HLT-NAACL, pages 272–280. The Association for Computational Linguistics.

Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. 2014. On the importance of text analysis for stock price prediction. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, pages 1170–1175. European Language Resources Association (ELRA).

Tim Loughran and Bill Mcdonald. 2011. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65.

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*, pages 3111–3119. Curran Associates, Inc.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327. Association for Computational Linguistics.

Clemens Nopp and Allan Hanbury. 2015. Detecting risks in the banking system by sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 591–600. Association for Computational Linguistics.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Alexander Dür, Linda Andersson, and Allan Hanbury. 2017. Volatility prediction using financial disclosures sentiments with word embedding-based IR models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1712–1721.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777. Association for Computational Linguistics.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 133–140, Stroudsburg, PA, USA. Association for Computational Linguistics.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *COLING*.

Paul C. Tetlock, Maytal Saar-tsechansky, and Sofus Macskassy. 2007. More than words: Quantifying language to measure firms ' fundamentals.

Christoph Kilian Theil, Sanja Stajner, and Heiner Stuckenschmidt. 2018. Word embeddings-based uncertainty detection in financial disclosures. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 32–37. Association for Computational Linguistics.

Ming-Feng Tsai and Chuan-Ju Wang. 2014. Financial keyword expansion via continuous word vector representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1453–1458. Association for Computational Linguistics.

Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan T. McDonald. 2010. The viability of web-derived polarity lexicons. In *HLT-NAACL*.

Iñaki San Vicente, Rodrigo Agerri, and German Rigau. 2014. Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages. In *EACL*.

Duy Tin Vo and Yue Zhang. 2016. Don't count, predict! an automatic approach to learning sentiment lexicons for short text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 219–224. Association for Computational Linguistics.

Chuan-Ju Wang, Ming-Feng Tsai, Tse Liu, and Chin-Ting Chang. 2013. Financial sentiment analysis for risk prediction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 802–808. Asian Federation of Natural Language Processing.

Leyi Wang and Rui Xia. 2017. Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 502–510. Association for Computational Linguistics.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohammadzaman Zamani and H Andrew Schwartz. 2017. Using twitter language to predict the real estate market. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 28–33.

A Appendix

A.1 Excess return regression results for multiple text variables

var	coeff	std coeff	t	R^2
$H4N_{RE}$	-0.88**	*-0.264*	**-2.19**	1.05
neg_{lm}	0.062	0.024	0.48	
$H4N_{RE}$	-0.739**	*-0.221*	**-2.23**	1.05
all_{lm}	-0.008	-0.008	-0.21	
$H4N_{RE}$	-0.836**	*-0.25*	**-2.15**	1.05
neg_unc_{lm}	0.027	0.016	0.28	
$H4N_{RE}$	-0.755**	*-0.226*	**-2.56**	1.05
neg_lit_{lm}	-0.003	-0.004	-0.12	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 10: This table shows results for regressions that combine $H4N_{RE}$ with single-feature manual L&M lists.

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
neg_{RE}	-0.37***	-0.111	**-2.96**	1.02
neg_{ADD}	-0.033	-0.0231	-1.03	1.00
neg_{lm}	-0.0607	-0.0242	-0.38	1.02
neg_{RE}	-0.274	-0.0822	-1.11	
neg_{RE}	-0.416***	*-0.124*	**-2.85**	1.02
neg_{ADD}	0.0298	0.0208	0.80	
neg_{lm}	-0.0421	-0.0168	-0.27	1.02
neg_{RE}	-0.346	-0.1037	-1.35	
neg_{ADD}	0.0277	0.0193	0.76	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 11: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the negative category.

var	coeff	std coeff	t	R^2
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
unc_{RE}	-0.377***	-0.075	**-2.77**	1.02
unc_{ADD}	0.0217	0.0065	0.21	1.00
unc_{lm}	0.209	0.0626	0.45	1.01
unc_{RE}	-0.668	*-0.133*	-1.05	
unc_{RE}	-0.643***	-0.128	**-3.14**	1.03
unc_{ADD}	0.198	0.0594	1.42	
unc_{lm}	-0.233	-0.0699	-0.42	1.03
unc_{RE}	-0.368	-0.0736	-0.54	
unc_{ADD}	0.234	0.0702	1.42	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 12: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the uncertain category.

var	coeff	std coeff	t	R^2
lit_{lm}	-0.0291	-0.026	-0.83	1.00
lit_{RE}	-0.056	-0.028	-0.55	1.02
lit_{ADD}	-0.0195	-0.0156	-0.70	1.00
lit_{lm}	-0.0759	*-0.0683*	-0.95	1.00
lit_{RE}	0.154	0.077	0.67	
lit_{RE}	-0.0261	-0.0130	-0.20	1.00
lit_{ADD}	-0.0136	-0.0108	-0.39	
lit_{lm}	-0.0753	-0.0677	-0.94	1.00
lit_{RE}	0.155	0.0775	0.66	
lit_{ADD}	-0.00107	-0.0008	-0.03	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 13: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the litigious category.

A.2 Volatility regression results for multiple text variables

var	coeff	std coeff	t	R^2
$H4N_{RE}$	0.748***	*0.224*	**4.44**	60.3
neg_{lm}	-0.096*	-0.038	**-2.55**	
$H4N_{RE}$	0.741***	*0.222*	**4.30**	60.3
all_{lm}	-0.0438**	-0.0481	**-2.95**	
$H4N_{RE}$	0.696***	*0.208*	**4.88**	60.3
neg_unc_{lm}	-0.054	-0.032	-1.86	
$H4N_{RE}$	0.693***	*0.207*	**4.24**	60.3
neg_lit_{lm}	-0.034**	-0.037	**-2.70**	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 14: This table shows results for regressions that combine $H4N_{RE}$ with single-feature manual L&M lists.

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
neg_{RE}	0.219***	0.0657	**3.57**	60.1
neg_{ADD}	0.032***	0.0224	**4.06**	60.0
neg_{lm}	0.0014	0.0005	0.02	60.1
neg_{RE}	0.217*	0.065	**1.96**	
neg_{RE}	0.233**	*0.0699*	**2.96**	60.1
neg_{ADD}	-0.0087	-0.006	-0.65	
neg_{lm}	0.00069	0.0002	0.01	60.1
neg_{RE}	0.232*	0.0696	**1.97**	
neg_{ADD}	-0.0087	-0.006	-0.66	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 15: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the negative category.

var	coeff	std coeff	t	R^2
unc_{lm}	0.119*	0.0356	**2.25**	60.0
unc_{RE}	0.167*	0.0334	**2.30**	60.0
unc_{ADD}	-0.013	-0.0039	-0.17	60.0
unc_{lm}	0.0432	0.012	0.28	60.0
unc_{RE}	0.112	0.0224	0.53	
unc_{RE}	0.222***	0.0444	**3.48**	60.1
unc_{ADD}	-0.088	-0.0263	-1.09	
unc_{lm}	0.151	*0.0453*	1.11	60.1
unc_{RE}	0.0419	0.0083	0.20	
unc_{ADD}	-0.111	-0.0332	-1.41	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 16: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the uncertain category.

var	coeff	std coeff	t	R^2
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
lit_{RE}	0.0080	0.004	0.20	60.0
lit_{ADD}	0.028	0.0224	1.07	60.0
lit_{lm}	-0.0635**	-0.057	**-2.93**	60.0
lit_{RE}	0.181*	*0.0905*	**2.46**	
lit_{RE}	-0.362	-0.181	-0.91	60.0
lit_{ADD}	0.041	0.0328	1.50	
lit_{lm}	-0.087***	-0.078	**-3.65**	60.1
lit_{RE}	0.174*	0.087	**2.42**	
lit_{ADD}	0.066*	0.0528	**2.23**	

*p $\leq$ 0.05, **p $\leq$ 0.01, ***p $\leq$ 0.001

Table 17: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the litigious category.