Automatic Detection of Speculation in Policy Statements

Sanja Štajner[1], Nicole Baerg[2], Simone Paolo Ponzetto[1] and Heiner Stuckenschmidt[1]
[1]Data and Web Science Research Group and [2]Department of Political Science
University of Mannheim, Germany
{sanja, simone, heiner}@informatik.uni-mannheim.de,
nicole.baerg@uni-mannheim.de

ABSTRACT

In this paper, we present the first study of automatic detection of speculative sentences in official monetary policy statements. We build two expert-annotated datasets. The first contains the transcripts of monetary policy meetings on the U.S. central bank's monetary policy committee (*Debates*). The second contains the official monetary policy statements (*Decisions*). We use the first part of the *Debates* dataset to build dictionaries with lexical triggers for *speculative* and *non-speculative* sentences. We then test their performance on an in-domain test set (the second part of the same dataset) and on an out-of-domain test set (the *Decisions* dataset) using several rule-based and machine learning classifiers. Our best classifiers achieve an accuracy of 82.5% (0.70 F-score on the *speculative* class), comparable with automatic detection of speculative sentences in Wikipedia articles.

Keywords

speculation detection, sentence classification, policy statements

1. INTRODUCTION

Before the 1990s, central banks around the world communicated their policy decisions and future policy plans opaquely, if at all. Since the 2000s, however, there has been a shift towards greater central bank transparency and more deliberate communication policy [5, 12]. In the case of the most powerful central bank in the world, the U.S. Federal Reserve Bank, policymakers spend an extraordinary amount of time deliberating the central bank's official policy statements. While scholars have paid increasing attention to central bank communications [4], thus far, most research focuses on using measures of textual complexity and readability/clarity [17, 9], or sentiment analysis [7].

The political economy literature highlights three main reasons for speculation in a political context. One, speculation may be willingly used in order to influence the behavior of other actors [28, 3]. Two, in a dynamic context, speculation can increase the flexibility of future policy actions [2, 24]. Three, speculation may also occur when verification of information is costly and/or being wrong means incurring reputation costs [8].

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Motivated by these studies, we present the first study on automatic detection of speculative sentences in this domain. We prepare two datasets – *Debates* (transcripts of the policy meetings) and *Decisions* (the official policy statements) – labelled for speculation at the sentence level by three expert annotators. We build several custom dictionaries which contain the most common *speculation* and *non-speculation* triggers (marked by the annotators) from the first part of the *Debates* dataset. Finally, we test the usefulness of the newly built dictionaries on the task of automatic detection of speculative sentences, using several rule-based and machine learning (ML) approaches, on two datasets – the remaining sentences from the *Debates* dataset (not used for dictionary extraction) and the entire *Decisions* dataset.

2. RELATED WORK

In the NLP community, automatic detection of speculative and non-speculative sentences has mainly focused on two types of texts, scientific texts from the (bio)medical domain [21, 23, 30, 13] and Wikipedia articles [14, 13].

In detecting speculative sentences in MEDLINE abstracts, Light *et al.* [21] find that a dictionary-based approach achieves higher accuracy (95%) than a Support Vector Machine (SVM) classifier, which uses term-based representation vectors (92%). The dictionary consists of 14 sub-strings, identified by the first author during manual annotation of the gene regulation abstracts. Medlock and Briscoe [23] use weakly supervised machine learning for automatic classification of speculative language in biomedical texts. Their best system achieves a 76% accuracy on the test set (on which human annotators achieved a κ=0.98 inter-annotator agreement[1]), while the sub-string matching classifier [21] achieves a 60% accuracy on the same test set. The best maximum entropy classifiers proposed by Szarvas [30] reach 0.85 and 0.82 F-measure (on the *speculative* class) on biomedical papers and medical reports, outperforming both previously proposed systems [21, 23]. This study was the first to allow speculation cues to be longer than one word, with one third of the features used in their best model being either bigrams or trigrams, and out of which half were phrases that had no unigram components of themselves in the feature set. Ganter and Strube [14] were the first to address the problem of automatic detection of speculative language in the broader domain of Wikipedia articles, using both corpus statistics and syntactic patterns.

The CoNLL-2010 shared task on learning to detect hedges and their scope in natural language text [13] was conducted on both previously mentioned domains. As can be expected – by comparing the inter-annotator agreement (IAA) on Wikipedia articles ($0.45 \leq \kappa \leq 0.80$ [14]) and the IAA on biomedical domain (κ =0.98 [23])

[1]See Cohen's Kappa inter-annotator agreement [10]

– the task of detecting speculation is significantly more difficult for Wikipedia articles than for the biomedical domain. The systems were evaluated using the F-score on the *speculative* class. The best system on the Wikipedia articles achieved a 0.60 F-score, while the best system on the biomedical domain achieved a 0.86 F-score.

These studies illustrate that automatic detection of speculative sentences is likely to require customisation for different domains. In addressing this task in political domain, we first build several dictionary-based approaches, following the procedures used by the above studies for the biomedical and Wikipedia domains [21, 14]. Taking into account that longer phrases are likely to be more suitable than unigrams for this task [30] and that syntactic patterns improve the performance of the systems [14], we do not restrict our dictionary entries to unigrams, and allow for words and word phrases to be of arbitrary lengths.

3. CORPUS CONSTRUCTION

We make use of two corpora, the corpus of internal deliberations by policymakers, reported in transcripts, in meetings on the U.S. central bank's monetary policy committee, the Federal Open Market Committee (FOMC) (*Debates*) and the corpus of the FOMC's official policy statements reported at the end of each meeting (*Decisions*).

3.1 Debates

The FOMC publishes verbatim transcripts of its policy meetings, and meeting transcripts are embargoed for five years. The FOMC meetings occur approximately every six weeks. We use the FOMC meeting transcripts between 2005 and 2008 ordered by meeting, speaker, and speech. We first randomly select 200 speeches (from the possible 428) with different speaker-meeting combination, and then chose a random selection of 300 sentences out of those 200 speeches. After discarding introductory sentences, such as "Thank you, Mr. Chairman.", and sentences that concerned meeting logistics, such as announcing or returning from lunch and coffee breaks and introductions, we were left with a final set of 292 sentences for annotation.

3.2 Decisions

The second corpora contains the FOMC's official statements. These policy statements represent official government public policy proclamations. These statements are important in that they influence financial market activity, such as market volatility in economic indicators, and shape expectations of future macroeconomic variables, such as unemployment levels, interest rates, and inflation rates. Starting in 2014, during the meetings, FOMC policymakers are offered four separate policy statement options. Policymakers must choose one of the four statements offered, either the status quo policy statement (the statement from the previous meeting), or an alternative statements from the remaining three. In order to decide which to accept, the statement is voted on and decisions are made by majority rule. Like the data from the FOMC meeting transcripts (*Debates*), there is ample variation in language used across sentences. Sometimes the policy sentence are more precise (i.e. *non-speculative*) whereas other times the policy sentences are more vague (i.e. *speculative*). We annotate all the official statements from 2004 (sentences which appear more than once are annotated only one time), irrespective of whether or not it was chosen, as this is the first year for which all possible statements are available.

3.3 Annotation Procedure

The annotation was performed by three annotators who had different characteristics that made for good subsets of annotator pairs. One pair had domain specific knowledge of central banking (annotators 1 & 3) whereas the second pair were native English speakers (annotators 2 & 3). The annotators marked each sentence for two items, using the annotation software MAE [29].

First, the annotators decided whether or not a given sentence was *speculative* or *non-speculative*. For example, when the statement was presented as a projection or a forecast, the sentence was coded as *speculative* and when the sentence was presented as a fact, the sentence was coded as *non-speculative*. Consider two examples that highlight the distinction between the two:

(1) **Speculative:** *Of course, even though recent core inflation data look pretty darn good, there may be forces at work that could undermine price stability.*

(2) **Non-speculative:** *Business sector demand is uniformly strong across small, medium, and large businesses.*

In addition to coding for speculation and non-speculation at the sentence level, the annotators also coded for words and word phrases that triggered their decision making (used later to build our dictionaries of *speculation* and *non-speculation* triggers). For example, the phrases *"I've become somewhat more"* and *"as suggesting"* were often marked by the annotators as the *speculation* triggers, whereas the phrases *"are consistent"* and *"continues to be"* were often coded as *non-speculation* triggers.

3.4 Inter-Annotator Agreement

A total of 292 sentences from the *Debates* dataset and 197 sentences from the *Decision* dataset were annotated by all three annotators. The pairwise inter-annotator agreement (IAA) for labeling sentences as *speculative* vs. *non-speculative* is shown in Table 1.

Annotators	Debates		Decisions	
	Agree	κ	Agree	κ
1 & 2	81.14%	0.62	84.77%	0.70
2 & 3	80.31%	0.58	73.60%	0.47
1 & 3	74.52%	0.49	82.74%	0.65
Average	78.66%	0.56	80.37%	0.61

Table 1: Pairwise inter-annotator agreement (Agree = percentage of cases in which both annotators assigned the same class; κ = Cohen's kappa)

The achieved pairwise IAA is comparable to that achieved on the Wikipedia dataset ($0.45 \leq \kappa \leq 0.80$) [14], suggesting that the tasks of marking *speculative* vs. *non-speculative* sentences in Wikipedia articles and in FOMC debates are equally difficult for humans (as opposed to a much easier task of marking *speculative* vs. *non-speculative* sentences in biomedical texts where human annotators achieved a $\kappa = 0.98$ [23]).

4. EXPERIMENTS

We divided *Debates* dataset into five smaller datasets:

- **Dict-L** – Randomly selected 132 sentences used to build a large dictionary of triggers for *speculative* and *non-speculative* sentences;
- **Dict-M** – Randomly selected 88 sentences from the *Dict-L* dataset, used to build a medium-size dictionary of triggers for *speculative* and *non-speculative* sentences;

- **Dict-S** – Randomly selected 44 sentences from the *Dict-M* dataset, used to build a small-size dictionary of triggers for *speculative* and *non-speculative* sentences;
- **TestDict/TrainML** – 80 sentences from the remaining set of sentences, with the constraint that of all of them were annotated with the same class by all three annotators (as this dataset was used for testing the usefulness of the dictionaries and for training ML classifiers, we wanted to assure that they contain only "easy", i.e. "clear" cases);
- **TestML** – The remaining 80 sentences (not necessarily annotated with the same class by all three annotators). The "gold standard" for this dataset was the majority class.

4.1 Dictionary Extraction

During the corpus annotation process, the annotators assigned a class (*speculative* or *non-speculative*) to each sentence and noted lexical triggers for each class. For *speculative* sentences, these were annotated as "positive" triggers and for *non-speculative* sentences, "negative" triggers. Using this information, we built one dictionary of triggers for *speculative* sentences and another for *non-speculative* sentences for each of the three datasets (*Dict-S*, *Dict-M*, *Dict-L*). We were particularly interested in whether the size of the dictionary (i.e. the number of annotated sentences for dictionary extraction) influenced performance.

Additionally, we built two dictionaries from existing sources, the *GeneralDict* and *FinanceDict*. The *GeneralDict* is a list of lexical triggers for *speculative* sentences comprised of a combination of detensifiers listed in [16, 18]. It consists of both unigrams and multi-word expressions. The *FinanceDict* is a list of words marked as the triggers of uncertainty in a large sample of 10,000 sentences from financial texts [22]. It only contains unigrams.

Our first goal was examining performance of the classification systems when exploiting domain-specific dictionaries (*Dict-S*, *Dict-M*, and *Dict-L*) and domain-independent dictionary (*GeneralDict*). Our second goal was investigating whether a dictionary that contained longer phrases (*Dict-S*, *Dict-M* or *Dict-L*) performs better than a dictionary that contains only unigrams (*FinanceDict*), such as what was found in the biomedical domain [30].

The number of entries for each dictionary is presented in Table 2. Several examples of lexical triggers for each type of sentences are also presented in Table 3.

Corpus	Speculative	Non-speculative
Dict-S	75	79
Dict-M	136	152
Dict-L	184	232
GeneralDict	33	NA
FinanceDict	297	NA

Table 2: The number of entries in each dictionary.

Speculative	Non-speculative
seems reasonably	*remained*
quite close	*continues to*
are likely to be	*held steady*
could have a	*continue to think*

Table 3: Examples of the dictionary entries.

4.2 Rule-Based Predictions

In our first set of experiments, we evaluated the usefulness of the dictionaries by building three simple classifiers based on the number of triggers for *speculative* and *non-speculative* sentences found in each of the 80 sentences from the TestDict/TrainML dataset (none of the test sentences appears in the datasets used for dictionary extraction):

- **Pred1** – If the number of triggers for *speculative* sentences is higher than the number of triggers for *non-speculative* sentences, then assign *speculative* class. If both are equal to '0', then assign *non-speculative* class.
- **Pred2** – If the number of triggers for *speculative* sentences is higher than the number of triggers for *non-speculative* sentences, then assign *speculative* class. If both are equal to '0', then assign *speculative* class.
- **Pred3** – If there are any triggers for *speculative* sentences found, then assign *speculative* class. Otherwise, assign *non-speculative* class.

All experiments were performed in two different settings:

1. Without any pre-processing (i.e. using string matching)
2. After stemming both dictionary entries and test sentences with the Porter stemmer [26] (NLTK[2] implementation).

4.3 Machine Learning Experiments

For each of the five dictionaries (*Dict-S*, *Dict-M*, *Dict-L*, *GeneralDict*, and *FinanceDict*), we built a feature set using all dictionary entries (in the case of *Dict-S*, *Dict-M*, and *Dict-L* we used both types of entries, *speculative* and *non-speculative*). Then, we performed two sets of experiments:

1. Training on the *TestDict/TrainML* dataset and testing on the *TestML* dataset;
2. Training on the *TestDict/TrainML* dataset and testing on the *Decisions* dataset.

For each sentence in the training and test set, we built a feature vector which contains the number of occurrences of each feature in the given sentence. The classifiers were built using nine machine learning algorithms implemented in the Weka Toolkit [15]: Logistic Regression [20], Naïve Bayes [19], Support Vector Machines [25] (with feature normalisation, with feature standardisation, and without any feature normalisation or standardisation), K-nearest neighbours [1], JRip – a propositional rule learner [11], J48 – C4.5 decision tree [27], and Random Forest [6]. Given that the Naïve Bayes algorithm achieved best results on a great majority of tasks, we present these results only.

5. RESULTS AND DISCUSSION

As seen in Table 4, on the *TestDict/TrainML* test set (a), the best accuracy is achieved for the *Pred3* rule-based approach using only the entries in the smallest of the three domain-specific dictionaries (*Dict-S*). It is interesting to note that stemming did not improve the accuracy of the systems. As expected, the domain-specific dictionaries (*Dict-S*, *Dict-M*, and *Dict-L*) are more suitable for this task, in these rule-based approaches, than the domain-independent (*GeneralDict*) or financial-domain (*FinancialDict*) dictionaries.

[2]http://www.nltk.org/

(a) Test dataset: TestDict/TrainML (majority class baseline = 63.75)

Method	Without stemming					With stemming				
	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict
Pred1	**71.25**	**67.50**	**72.50**	NA	NA	**67.50**	**66.25**	**70.00**	NA	NA
Pred2	50.00	63.75	**66.25**	NA	NA	55.00	**68.75**	63.75	NA	NA
Pred3	***76.25***	**70.00**	63.75	61.25	36.25	***76.25***	**70.00**	62.50	56.25	36.25

(b) Test dataset: TestML (majority class baseline = 68.75)

Method	Without stemming					With stemming				
	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict
Pred1	61.25	62.50	62.50	NA	NA	62.50	57.50	58.75	NA	NA
Pred2	40.00	50.00	45.00	NA	NA	41.25	47.50	41.25	NA	NA
Pred3	60.00	56.25	48.75	61.25	31.25	60.00	58.75	48.75	58.75	31.25
Naïve Bayes	**76.25**	**80.00**	***82.50***	65.00	**77.50**	**71.25**	**77.50**	**77.50**	65.00	***82.50***

(c) Test dataset: Decisions (majority class baseline = 45.18)

Method	Without stemming					With stemming				
	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict	Dict-S	Dict-M	Dict-L	GeneralDict	FinanceDict
Pred1	**56.85**	**55.33**	**58.88**	NA	NA	**55.33**	**56.35**	**60.41**	NA	NA
Pred2	**46.70**	44.16	**52.28**	NA	NA	45.18	39.09	**46.19**	NA	NA
Pred3	**58.37**	**59.39**	**58.38**	**60.91**	**54.82**	**56.35**	**50.76**	**53.30**	**64.47**	**54.82**
Naïve Bayes	**58.88**	**60.41**	**67.51**	**50.25**	**70.05**	**60.41**	**64.47**	**68.02**	**55.33**	***71.06***

Table 4: Accuracy (i.e. the percentage of correctly classified instances) on different test datasets. The results which outperform the majority class baseline (which always assigns the majority class of the training dataset, i.e. the non-speculative class) are presented in bold. The highest achieved scores on each test dataset are marked between two '*'

On the *TestML* test set (b), only the Naïve Bayes classifier outperformed the majority class baseline. The best results were achieved when the feature set consisted either of the entries of the *Dict-L* (without stemming) or the entries of the *FinanceDict* (with stemming).

On the out-of-domain (*Decisions*) test set (c), the Naïve Bayes classifier trained on the *FinanceDict* feature set (with stemming) again achieved the best accuracy (though lower than on the in-domain test set), while the *Dict-L* feature set (without stemming) achieved a slightly lower accuracy. In the *Pred3* rule-based approach on the *Decisions* test set, the domain-independent *General-Dict* feature set led to better results than the feature sets built using the domain-specific dictionaries (*Dict-S*, *Dict-M*, *Dict-L*) or the *FinanceDict*. These results suggests that the newly built domain-specific dictionaries lead to overfitting on the in-domain datasets.

5.1 Comparison with the State of the Art

Our best classifiers on the in-domain (*TestML*) test set (Naïve Bayes using *Dict-L* without stemming or *FinanceDict* with stemming) and the out-of-domain (*Decisions*) test set (Naïve Bayes using *FinanceDict* with stemming) achieved a 0.70 F-scores on the *speculative* class. Although the systems are not directly comparable to those of the CoNLL-2010 shared task [13] due to different domains, training and test sets, it is interesting to note that our simple, dictionary-based systems achieve significantly higher F-scores for the *speculative* class (0.70) than the best shared-task system on the Wikipedia domain (0.60). Given the similar scores for IAA for our domain and for Wikipedia articles [23], we expect that both tasks are similarly difficult.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented several systems for automatic detection of speculative sentences in central bank's policy statements. Using two datasets annotated at a sentence-level by three expert annotators, we built several dictionaries of lexical triggers for *speculative* and *non-speculative* sentence detection.

Our results showed that the use of the Naïve Bayes classifier on our domain-specific dictionaries and the finance-domain dictionary perform the best on both in-domain and out-of-domain test sets. In the rule-based approaches, however, the domain-independent dictionary performs better.

Finally, our best classification systems, the Naïve Bayes algorithms using the entries from the biggest domain-specific dictionary and the finance-domain dictionary as features, achieve results comparable to the state-of-the-art systems trained and tested on Wikipedia.

In future research, we would like to experiment with a richer set of features, combining both lexical and syntactic features.

Acknowledgements

The authors acknowledge support for this work by the SFB 884 on the "Political Economy of Reforms" at the University of Mannheim (project C4), funded by the German Research Foundation (DFG), and also wish to thank to the anonymous reviewers for their constructive and helpful comments.

7. REFERENCES

[1] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] A. Alesina and A. Cukierman. The Politics of Ambiguity. *The Quarterly Journal of Economics*, 105, 1990.

[3] N. Baerg and C. Krainin. Committees and Distortionary Vagueness. Working paper, 2016.

[4] D. Bholat, S. Hansen, P. Santos, and C. Schonhardt-Bailey. Text mining for central banks: Handbook. *Centre for Central Banking Studies*, (33):1–19, 2015.

[5] A. S. Blinder, M. Ehrmann, M. Fratzscher, J. De Haan, and D.-J. Jansen. Central bank communication and monetary policy: A survey of theory and evidence. Technical report, National Bureau of Economic Research, 2008.

[6] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[7] S. Cannon. Sentiment of the FOMC: Unscripted. 2015.

[8] G. Chortareas, D. Stasavage, and G. Sterne. Does it pay to be transparent? International evidence from central bank forecasts. *Review - Federal Reserve Bank of Saint Louis*, 84(4):99–118, 2002.

[9] M. Cihák, D.-J. Jansen, and A. Bulir. Clarity of central bank communication about inflation. *IMF Working Paper*, 2012.

[10] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

[11] W. W. Cohen. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

[12] N. N. Dincer, B. Eichengreen, et al. Central Bank Transparency and Independence: Updates and New Measures. *International Journal of Central Banking*, 10(1):189–259, 2014.

[13] R. Farkas, V. Vincze, G. Móra, J. Csirik, and G. Szarvas. The CoNLL-2010 Shared Task: Learning to Detect Hedges and Their Scope in Natural Language Text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (Shared Task)*, pages 1–12. ACL, 2010.

[14] V. Ganter and M. Strube. Finding Hedges by Chasing Weasels: Hedge Detection Using Wikipedia Tags and Shallow Linguistic Features. In *Proceedings of the ACL-IJCNLP Conference (Short Papers)*, pages 173–176. ACL, 2009.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[16] A. Hübler. *Understatements and hedges in English*. Amsterdam:Philadelphia: J. Benjamins, 1983.

[17] D.-J. Jansen. Does the Clarity of Central Bank Communication Affect Volatility in Financial Markets? Evidence from Humphrey-Hawkins Testimonies. *Contemporary Economic Policy*, 29(4):494–509, 2011.

[18] G. Jason. Hedging as a Fallacy of Language. *Informal Logic*, 1988.

[19] G. H. John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[20] S. le Cessie and J. van Houwelingen. Ridge Estimators in Logistic Regression. *Applied Statistics*, 41(1):191–201, 1992.

[21] M. Light, X. Y. Qiu, and P. Srinivasan. The Language of Bioscience: Facts, Speculations, and Statements in Between. In *Proceedings of the HLT-NAACL Workshop on Linking Biological Literature, Ontologies and Databases (Biolink)*, pages 17–24, 2004.

[22] T. Loughran and B. McDonald. When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance*, 2010.

[23] B. Medlock and T. Briscoe. Weakly Supervised Learning for Hedge Classification in Scientific Literature. In *Proceedings of teh 45th Annual Meeting of the Association of Computational Linguistics*, pages 992—999, 2007.

[24] A. Meirowitz. Informational Party Primaries and Strategic Ambiguity. *Journal of Theoretical Politics*, 17(1):107–136, 2005.

[25] J. C. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods – Support Vector Learning*. 1998.

[26] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[27] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[28] J. K. Staton and G. Vanberg. The value of vagueness: delegation, defiance, and judicial opinions. *American Journal of Political Science*, 52(3):504–519, 2008.

[29] A. Stubbs. MAE and MAI: lightweight annotation and adjudication tools. In *Proceedings of the 5th Linguistic Annotation Workshop at ACL*, 2011.

[30] G. Szarvas. Hedge Classification in Biomedical Texts with a Weakly Supervised Selection of Keywords. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 281–289, 2008.