Graph Structured Semantic Representation and Learning for Financial News

Boyi Xie and **Rebecca J. Passonneau**
Center for Computational Learning Systems
Columbia University
New York, New York 10115
{xie@cs.|becky@ccls.}columbia.edu

Abstract

This study links stock prices of publicly traded companies with online financial news to predict direction of stock price change. Previous work shows this to be an extremely challenging problem. We develop a very high-dimensional representation for news about companies that encodes lexical, syntactic and frame semantic information in graphs. Use of a graph kernel to efficiently compare subgraphs for machine learning provides a uniform feature engineering framework that integrates semantic frames in document representation. Evaluated on a news web archive against two benchmarks, only our approach beats the majority class baseline, and with statistically significant results.

Introduction

Graphs are a flexible and efficient data structure for problems as diverse as prediction of toxicity based on molecular structure (Wale, Watson, and Karypis 2008), analysis of virtual 3-D scenes (Fisher, Savva, and Hanrahan 2011), and social network analysis (Ediger et al. 2010). They have been used in many NLP tasks, such as polarity of words (Hassan and Radev 2010), dependency parsing (McDonald et al. 2005), an abstract meaning representation (Banarescu et al. 2013) for PropBank-style semantic analysis. Little if any work applies graphs to document representation. The thesis of our study is that in a noisy domain such as finance, large-scale data analytics benefits from a linguistically rich representation that can support a range of features based on words, syntactic relations, and frame semantics (semantic roles and concepts). We test this thesis on seven years of financial news to predict direction of stock price change for over four hundred publicly traded companies. Our graph-based document representation exhibits superior performance over the majority class baseline and two existing benchmarks based on vector and tree representations.

Entity-driven text analytics is a recent area of active research where large collections of documents are analyzed to study entity mentions in text, often to predict real world outcomes of the entities. For example, O'Connor, Stewart, and Smith (2013) associate newswire with political information to learn international relations among countries. Our work also relies on news articles to make predictions about a specific class of entities: publicly-traded companies. *SemGraph* is a representation we propose for entity-driven text analytics. It provides a very high-dimensional feature space that encodes lexical, syntactic, and frame semantic information in graphs. The input to *SemGraphs* consists of sentences that mention the relevant companies after they have been parsed into dependency trees, and then into semantic frame parses (Das et al. 2010). Semantic frames capture role relations, and abstract concepts (Fillmore 1976). The graph kernel we use for machine learning mines substructures of the graph to detect predictive features. Our previous work proposed a hybrid vector and tree space (*SemTreeFWD*) that also used semantic frames as input (Xie et al. 2013). It artificially constrained the depth of paths in the tree, and supported a much more limited range of features. When combined with *BOW* features, it had only modest performance on the stock prediction task. *SemGraph*, a much more general representation, outperforms both *BOW* and *SemTreeFWD*.

Related Work

Text mining in the financial domain has a growing presence (Tetlock 2007; Engelberg and Parsons 2011). Kogan et al. (2009) analyzed quarterly earning reports to predict stock volatility and to predict delisting of companies. Luss and d'Aspremont (2008) used text classification to model price volatility. Their representation is a vector that merges bag-of-words with equity returns features. Feldman et al. (2011) use an information extraction approach combined with a sentiment lexicon to model daily price movements that requires extensive manual engineering. Our work leads to rich features without the need for manual feature engineering.

Past work on the benefits of structured representations in machine learning for NLP has focused mainly on tree kernel learning (Collins and Duffy 2002; Moschitti 2006). Graph kernels for machine learning can be categorized into three classes: graph kernels based on walks (Kashima, Tsuda, and Inokuchi 2003) and paths (Borgwardt and Kriegel 2005), graph kernels based on limited-size subgraphs (Horváth, Gärtner, and Wrobel 2004), and graph kernels based on subtree-like patterns (Mahé and Vert 2009). By analogy with tree-kernels for NLP, we select a graph kernel, Weisfeiler-Lehman (Shervashidze et al. 2011), that measures similarity of subtree-like patterns.

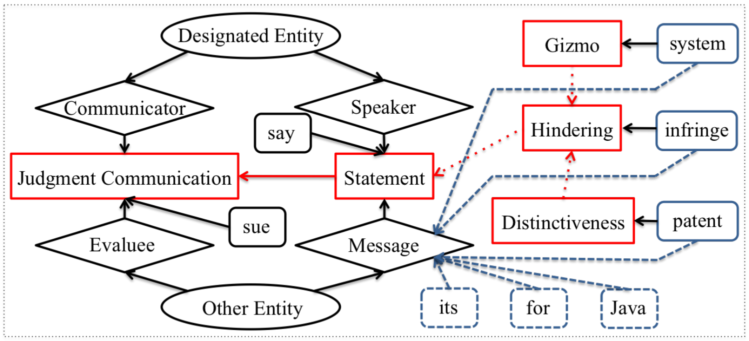

Figure 1: *SemGraph* for: *Oracle sued Google, saying Google's Android system infringes its patents for Java. SemGraph* incorporates the roles of the entity, semantic frame attributes, dependency structures, and lexical items.

SemGraph Construction

SemGraph aims at a concise representation of lexical, syntactic and semantic information. Our design criteria were to 1) focus on entities of interest, 2) facilitate feature engineering for diverse features, 3) capture relational information, and 4) avoid topological limitations (e.g., path length; distinction between terminal and non-terminal nodes). For the sentence in Figure 1, for example, our goal is to capture the distinctive words (e.g., *sue*), and the relation between *Oracle* and *Google* in a way that distinguishes their semantic roles (e.g., *complainant, plaintiff*).

Figure 1 shows an example *SemGraph* for a sentence with two semantic frames (Judgment Communication and Statement). The nodes are frame names (boxes), frame elements–roles (diamonds), frame targets–lexical items that trigger frames (rounded boxes), company entities (ellipses), and lexical items (dashed rounded boxes). An edge connects a pair of nodes in the following three cases: (1) a frame target and the frame it evokes (e.g. ⟨sue, Judgment_communication⟩); (2) a frame element and the frame it belongs to (e.g. ⟨Communicator, Judgment_communication⟩); and (3) a designated entity (DE), or other entity (OE), and the frame element (or semantic role) it fills (e.g. ⟨DE, Communicator⟩).

Features Beyond Semantic Frames As shown in Figure 1, *SemGraph* consists of nodes of designated entities (DE), frames (F), frame targets (FT) and frame elements (FE), plus orange edges for relations of ⟨DE, FE⟩, blue edges for ⟨FT, F⟩, and green edges for ⟨FE, F⟩. Notice that vanilla *SemGraph* contains isolated subgraphs if the frames don't have common entity mentions. Interestingly, we can model relations among frames through dependency parsing. Dependencies among frames (red edges - both solid and dotted) can be incorporated, denoted by *SemGraphDep*. To reduce the size of the graph, subgraphs of frames that are along the dependency path from the *ROOT* of the dependency parse up to the frames that have a DE mention are retained, others (i.e. nodes along the dotted red edges) are excluded. Furthermore, lexical items can be attached to the frame elements that they belong to (i.e. the dashed nodes and edges). We denoted it *SemGraphDepW*, which forms a unified representation to model the roles of designated entities, semantic frames, dependency relations and lexical items.

Directed SemGraphs exist for all the *SemGraph* variations listed above, where all edges become directed. The direction of an edge is determined by syntactic dependency as follows: 1) the frame in a subordinate clause depends on a frame in its superordinate clause; 2) frame elements depend on their frames; 3) words that fill a frame element depend on the frame element; 4) designated entity nodes depend on the frame element where they are the role filler.

Graph Kernel Learning for *SemGraph*s

Among a variety of graph kernels as described in the Related Work Section, we selected the Weisfeiler-Lehman (WL) graph kernel (Shervashidze et al. 2011) for SVM learning and feature exploration. It can measure similarity between graphs with respect to different neighborhood sizes specified by the user. This allows us to test many classes of *SemGraph* features for minimal engineering costs. It also has a lower computational complexity compared to other graph kernels.

The WL kernel computation is based on the Weisfeiler-Lehman test of isomorphism (Weisfeiler and Lehman 1968), which iteratively augments the node labels by the sorted set of their neighboring node labels, and compress them into new short labels, called multiset-labels. WL graph kernel applies the idea of neighbor augmentation to iteratively measure the similarity between graphs using dynamic programming. The kernel computation takes into account different levels of the node-labelings, from the original labelings to increasingly large h-degree neighborhoods (stepsizes). The full kernel for a given h is then the sum of the kernel computations for each stepsize from 0 to h.

Experiments

Reuters news data from 2007 to 2013 that covers eight GICS[1] sectors are used in our experiments.[2] A data instance is all the news associated to a company on a day, for companies whose price changed above a threshold between the

[1] Global Industry Classification Standard.

[2] Two sectors are not included in this study. The Financial sector is usually excluded from financial analytics. Telecommunicatons has insufficient data due to few companies that remain in the S&P 500 in our time frame.

Ticker	Baseline	Vanilla SemGraph				SemGraphDep				SemGraphW				SemGraphDepW			
		h=0	h=1	h=2	h=3	h=0	h=1	h=2	h=3	h=0	h=1	h=2	h=3	h=0	h=1	h=2	h=3
BHI	53.01	48.93	55.25	**56.28**	55.77	48.93	**50.58**	50.41	50.58	50.91	51.90	**52.56**	52.23	50.91	**52.23**	52.23	51.07
COP	53.20	53.31	55.99	56.16	**56.98**	53.31	56.31	56.62	**57.10**	51.58	54.73	54.73	**55.05**	51.58	54.73	55.05	**56.62**
CVX	50.22	52.92	54.61	57.24	**57.68**	52.92	**54.86**	53.78	54.64	51.19	**52.48**	51.84	51.40	**51.19**	50.97	50.76	50.54
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
OXY	55.52	53.48	**53.63**	48.90	49.21	53.48	54.04	54.04	**55.99**	54.87	53.76	**55.15**	55.15	54.87	54.32	**56.82**	56.82
VLO	53.99	51.14	**53.32**	49.67	48.51	51.14	55.08	**55.54**	54.93	50.99	**54.32**	53.57	53.26	50.99	**54.32**	54.02	53.57

Table 1: A breakdown of performance by stepsizes (h) using WL graph kernels for 4 variants of *SemGraph*. It shows the leave-one-out accuracies for some sample companies in the Energy sector.

Sector	Baseline	Benchmarks		SemGraphs				
		BOW	SemTreeFWD	directed	Vanilla SemGraph	SemGraphDep	SemGraphW	SemGraphDepW
Energy	53.95±3.36	52.56±3.97	53.53±4.84	n	54.94±5.71	**55.93**±5.60*	55.71±6.16*	55.80±5.89*
				y	54.69±5.09	55.61±5.50	55.10±5.80	55.11±6.30
Materials	55.00±2.88	53.18±5.23	52.73±5.60	n	56.20±4.07	55.08±4.47	55.34±5.66	55.28±5.69
				y	**57.07**±4.30*	55.29±4.09	55.10±6.42	55.11±6.30
Industrials	54.25±3.85	52.89±5.91	52.90±5.21	n	55.24±6.21	54.96±5.54	54.35±5.15	54.41±5.36
				y	54.94±6.17	**55.32**±5.41	53.94±5.36	54.58±5.18
Consumer Discretionary	54.32±4.18	53.91±4.73	54.09±5.86	n	54.52±5.96	55.66±7.19	55.56±4.28*	**55.67**±4.41*
				y	54.27±6.26	55.46±5.66	55.36±4.58*	55.40±4.77*
Consumer Staples	54.85±3.24	52.82±4.07	53.78±3.76	n	55.74±4.98	53.97±3.88	53.14±6.06	53.21±5.71
				y	**56.13**±4.95	54.35±3.39	52.67±5.54	52.83±5.63
Health Care	56.44±4.51	52.75±3.86	54.31±6.46	n	55.61±6.62	**57.89**±5.39	55.32±5.31	55.29±5.39
				y	55.51±6.70	57.55±5.54	55.33±5.49	55.48±5.38
Information Technology	53.95±4.07	52.42±3.64	52.79±6.84	n	**56.18**±4.50*	54.07±5.03	53.81±6.06	53.76±6.15
				y	55.59±4.16*	54.10±3.98	53.55±6.20	53.77±5.91
Utilities	53.82±2.75	51.66±4.24	51.75±5.23	n	54.87±6.28	54.03±4.53	**55.16**±5.62	55.00±5.48
				y	54.10±5.47	54.59±5.08	55.01±5.51	54.68±5.33

Table 2: The means and standard deviations of the leave-one-out accuracy over the companies in each of the eight GICS sectors. The performance of all variations of *SemGraph* are shown. Boldface values are the best performance across different *SemGraph*s. $*$ indicates a p-value$<$.05 compared to baseline.

closing price on the day of the news and the closing price on the following day. In this experiment, we use a threshold of 2% that corresponds to a moderate fluctuation. A binary class label {-1, +1} indicates the direction of price change on the next day after the news associated to the data instance. Two benchmarks are compared: *BOW* - a bag-of-words model that consists of unigrams, bigrams, and trigrams trained with linear SVM, and *SemTreeFWD* (Xie et al. 2013) - a vector/tree space hybrid that contains semantic frames, lexical items, and psycholinguistic dictionary-based features trained with Tree Kernel SVM (Moschitti 2006).

Two phases of experiments apply *SemGraph* to large-scale semantic analysis. The accuracy of leave-one-out cross-validation is reported. An advantage of the WL kernel is that it facilitates exploration of user-specified neighborhood sizes in a graph. Therefore, Phase I tests the efficacy of different stepsizes for each of the eight *SemGraph* variants. Stepsizes from 0 to 3 are used here to provide a more direct comparison to the *SemTree* representation of Xie et al. (2013). Paths from the root node of *SemTree* loosely correspond to paths of up to stepsize 3 from the designated entity nodes in vanilla *SemGraph*. The WL kernel, however, considers the neighborhood three steps from all nodes, not just the designated entity nodes.

Table 1 presents the Phase I results on impact of stepsize. The undirected versions of the 4 variants of *SemGraph* are shown here for a few companies from the *Energy* sector. Numbers in boldface identify the best performing stepsize for each *SemGraph* variant; the underlined values are the best performance across all variants for a given company. No single stepsize consistently performs best across companies, or across *SemGraph* variants. In a majority of cases there is improvement after including at least 1-degree neighbors, and sometimes the best performance is at h=2 or h=3. For example, the best performance for *ConocoPhillips* (*COP*) for each variant uses h=3, and *SemGraphDep* performs best among the 4 variants of *SemGraph*. In a statistical test for all companies in this sector, including at least 1-degree neighbors performs significantly better than using only 0-degree neighbors.

The Phase II experiments assess average performance across all companies in a sector, using the best stepsize identified for each pairing of a *SemGraph* with a given company from Phase I. Here the goal is to identify which configurations of *SemGraph* perform best across an entire sector. Table 2 presents Phase II results. Numbers in boldface identify which of the 8 variants has the best mean accuracy for the sector. T-tests that compare the means of each *SemGraph* to the baseline indicate that in 4 out of 8 sectors, one or more *SemGraph* variants have significantly better accuracy than the baseline. The benchmarks, *BOW* and *SemTreeFWD*, never beat the baseline. With a higher accuracy in the majority of cases, no single variation of *SemGraph* consistently significantly outperforms the baseline.

More expressive representation does not always lead to higher accuracy. Including syntactic dependency information alone (*SemGraphDep*) helps more often than including lexical information alone (*SemGraphW*): *SemGraphDep* leads to higher mean accuracy in 3 of the 8 sectors, while *SemGraphW* accuracy is superior in only 1 of the sectors. T-tests to compare mean accuracy of *SemGraphDep* and *SemGraphW* indicate that *SemGraphDep* is significantly better in 4 of the 16 cases (8 sectors, directed versus undirected graphs), and *SemGraphW* is never significantly better than *SemGraphDep*. The directed versions of *SemGraph* achieve the best mean accuracy in 3 of 8 sectors (Materials, Industrials, and Consumer Staples), and in one of these cases (Materials), the difference is statistically significant. One advantage to the directed versions is efficiency. They reduce the kernel computation asymptotically by a half, since they only allow the flow of information to pass edges through one direction.

Conclusion

In this study, we link S&P 500 companies with the Reuters news archive to mine the impact of financial news on stock price prediction. We presented a novel graph-structured semantic representation (*SemGraph*) with WL graph kernel learning. The advantages of *SemGraph* stem from the use of semantic frame features to generalize word meanings in a flexible and extensible graph structure, where atomic and relational linguistic information can be modeled and learned. In the financial industry, each sector is viewed as a distinct semantic domain. Feature analysis also demonstrates the contribution of graph-structured relations that encompass a variety of linguistic features: lexical items, syntactic dependencies, and frame semantics. The patterns discovered are well beyond the expressiveness of conventional methods. A direction for future research is to investigate a weighting scheme to distinguish node and edge feature types in *SemGraph*s, and additional enrichments of the representation, for example, with sentiment lexicons or vocabulary distributions learned through latent Dirichlet allocation. Another possibility includes domain adaptation to facilitate prediction across companies and sectors.

References

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract meaning representation for sembanking.

Borgwardt, K. M., and Kriegel, H.-P. 2005. Shortest-path kernels on graphs. In *Proceedings of ICDM05*, 74–81. IEEE.

Collins, M., and Duffy, N. 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL02*, 263–270. ACL.

Das, D.; Schneider, N.; Chen, D.; and Smith, N. A. 2010. Probabilistic frame-semantic parsing. In *Proc. of NAACL-HLT*.

Ediger, D.; Jiang, K.; Riedy, J.; Bader, D. A.; and Corley, C. 2010. Massive social network analysis: Mining twitter for social good. In *ICPP '10*, 583–593. Washington, DC: IEEE Computer Society.

Engelberg, J., and Parsons, C. A. 2011. The causal impact of media in financial markets. *Journal of Finance* 66(1):67–97.

Feldman, R.; Rosenfeld, B.; Bar-Haim, R.; and Fresko, M. 2011. The stock sonar - sentiment analysis of stocks based on a hybrid approach. In *Proceedings of IAAI11*.

Fillmore, C. J. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences* 280(1):20–32.

Fisher, M.; Savva, M.; and Hanrahan, P. 2011. Characterizing structural relationships in scenes using graph kernels. In *SIGGRAPH11*, 34:1–34:12. New York, NY, USA: ACM.

Hassan, A., and Radev, D. 2010. Identifying text polarity using random walks. In *Proceedings of ACL10*, 395–403. ACL.

Horváth, T.; Gärtner, T.; and Wrobel, S. 2004. Cyclic pattern kernels for predictive graph mining. In *Proceedings of KDD*, 158–167.

Kashima, H.; Tsuda, K.; and Inokuchi, A. 2003. Marginalized kernels between labeled graphs. In *ICML03*, 321–328.

Kogan, S.; Levin, D.; Routledge, B. R.; Sagi, J. S.; and Smith, N. A. 2009. Predicting risk from financial reports with regression. In *Proceedings of NAACL09*, 272–280. ACL.

Luss, R., and d'Aspremont, A. 2008. Predicting abnormal returns from news using text classification. *CoRR* abs/0809.2792.

Mahé, P., and Vert, J.-P. 2009. Graph kernels based on tree patterns for molecules. *JMLR* 75(1):3–35.

McDonald, R.; Pereira, F.; Ribarov, K.; and Hajič, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of EMNLP05*, 523–530. ACL.

Moschitti, A. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL06*.

O'Connor, B.; Stewart, B. M.; and Smith, N. A. 2013. Learning to extract international relations from political context. In *Proceedings of ACL13*, 1094–1104. Sofia, Bulgaria: ACL.

Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *JMLR* 12:2539–2561.

Tetlock, P. C. 2007. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*.

Wale, N.; Watson, I.; and Karypis, G. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems* 14(3):347–375.

Weisfeiler, B., and Lehman, A. A. 1968. A reduction of graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya, Ser. 2, no. 9*.

Xie, B.; Passonneau, R. J.; Wu, L.; and Creamer, G. 2013. Semantic frames to predict stock price movement. In *Proceedings of ACL13*, 873–883. Sofia, Bulgaria: ACL.