Mapping Deep NLP to Knowledge Graphs: An Enhanced Approach to Analyzing Corporate Filings with Regulators

Damir Cavar, Matthew Josefy
Indiana University
Bloomington, IN
{dcavar,mjosefy}@iu.edu

Abstract

Filings submitted by companies to the Securities and Exchange Commission provide a tremendous corpus for application of advanced natural language processing techniques. While business scholars actively utilize these texts, interdisciplinary efforts hold substantial promise to advance knowledge and techniques. In this study, we utilize deep natural language processing techniques to extract meaningful knowledge from SEC filings. We construct a comprehensive pipeline that extracts the original filings, processes them in order to recognize component segments for distinct analysis, feeds each text through multiple NLP processors to obtain optimal recognition of the linguistic properties, and ultimately seeks to construct a comprehensive knowledge graph of how companies, their executives and their directors are linked to one another, or how various risks are identified, weighted, and handled over long periods of time. We thus link advanced NLP techniques and knowledge graphing approaches, contributing greater domain specific knowledge to advance linguistic approaches and potentially discovering underlying networks that would be difficult to detect with other approaches.

Keywords: Deep NLP, Topic Modeling, Knowledge Graphs

1. Introduction

We describe our research activities in the domain of engineering and tuning Natural Language Processing (NLP) technologies for the analysis of business reports. The goal of this project is to provide data and detailed analyses of corporate reporting, focusing initially on the Securities and Exchange Commission (SEC) reports, while constructing our pipeline and process in such a way as to deploy in future to also analyze international financial reporting sources in similar fashion.

The SEC is the body in the U.S. charged with ensuring fair and transparent markets. To that end, the SEC requires publicly-listed firms in the U.S. (and other firms who meet certain thresholds) to comply with disclosure requirements. These reports become publicly-available through SEC EDGAR (`https://www.sec.gov/edgar.shtml`) which serves as a repository of over 21 million corporate filings. Information within these reports is often market-sensitive. Indeed, recent research found that certain subscribers who had access to the SEC's public dissemination system (PDS) were able to profit merely by having access to information 30 seconds before it was made available to the public on EDGAR (Jonathan L. Rogers and Zechman, 2017).

The SEC was founded in 1933 and its electronic repository dates back to 1984. Accordingly, longitudinal data is available dating back decades, allowing the potential to study networks of firms and the executives and board members associated with them. Thus, in our current work, one key aim is to map management and board members of firms in a network of relations over time. Thus our project uniquely integrates aspects of deep processing to harvest information with advanced network maps to ascertain previously obscured relationships between focal nodes, which in this case may be either individuals or companies. The filings contain detailed descriptions of individuals involved in the firms in specific roles, as well as the description of corporate relations between firms. Mapping of individuals and their roles and relations to other individuals and institutions over time allows us to study relations using complex network analyses (Newman et al., 2006). Ultimately, these networks may be useful in understanding the survival and failure of firms (Josefy et al., 2017).

Besides network relations between individuals over time, a specific interest area is the analysis and mapping of perceived risks and the corresponding risk management strategies, also along the time axis, as reported by the firms across different business types. The annual reports of firms filed on Form 10-K contain a detailed picture of the risks a firm faces at the time of the reporting. An analysis and classification of the particular risks and the arrangement along the time axis provide an extremely valuable analytical instrument for the understanding of the evolution of risks and risk management strategies in different industry sectors, as well as a correlation with national and geopolitical historical events and developments.

These main interest areas motivated our approach to utilize advanced Natural Language Processing (NLP) and Topic Modeling (Blei, 2012) methodologies for the analysis of the different types of SEC filings.

The filings are available in common accessible formats, e.g. raw HTML or the standardized eXtensible Business Reporting Language (XBRL) (Engel et al., 2013) XML format. XBRL is an XML language that is freely available as a global standard for exchanging business information. Independent of the formats, the document content is arranged in ways specific to the filing institution, and not in a strictly predefined order. This requires either manual processing and annotation of the content by section and paragraph, or automatic classification tools have to be developed that detect sections by content.

The lack of a uniform structure and clear semantic content specification in the filings implies that the target content (e.g. director biographies, work histories and board description, risk management sections) needs to be parsed and analyzed using Machine Learning, Document Classification, and Natural Language Processing (NLP) technologies. On the one hand, the relevant sections and paragraphs have to be identified. On the other hand it is necessary to map parsed and identified concepts, relations and other properties of concepts, and other semantic properties of the content and meta-information onto a structured data representation for subsequent analysis and processing. Common NLP technologies that are freely accessible come with serious limitations and unacceptable error rates when applied to complex business language in the specific document types. We developed various NLP components and pipelines to remedy this.

In the following we describe in more detail our approach to the problem of the identification of target content in semi-structured business reports of the SEC type, as well as the architecture and technologies that we developed and utilized to maximize the quality of the NLP results for knowledge mapping to graph representations and deep linguistic topic modeling.

2. Previous Work

When looking at previous work in the domain of interest, we should differentiate between a.) the analysis of SEC reports using NLP, b.) specific NLP technologies tuned for the particular task and business language, c.) processing of the particular information (e.g. extraction of relations between individuals and institutions mapped to time, analysis of risks and risk management over time), and d.) mapping the resulting information on knowledge graphs and advanced knowledge information systems.

While we are more than certain that many business make use of NLP technologies for processing of business documentation, financial reports, and other open or closed reports by firms, we can only comment on information accessible to us in form of publications or free and open toolkits and technologies.[1]

While there are commercial APIs that provide access to the SEC filings, these filings are also publicly available to any user. Therefore, we will not discuss details of such commercial systems here. There are numerous open modules and toolkits to process and access the SEC EDGAR data, a search online will reveal all the relevant sites and information. We will not go into detail here. There are a few resources and APIs that allow access to the SEC EDGAR repository or analyses of the bulk data.

The *CorpWatch API* (`http://api.corpwatch.org/`), for example, "uses automated parsers to extract the subsidiary relationship information from Exhibit 21 of companies' 10-K filings." There are various Python modules that facilitate crawling and downloading of the filings, as for example the Python module SEC-Edgar, which implements a Sphinx crawler.

Arelle (`arelle.org`) (Fischer and Mueller, 2011b; Fischer and Mueller, 2011a; Fischer, 2013) is a large project that provides a set of tools and applications geared for the analysis and processing of the XBRL format of the SEC EDGAR reports (and other filings using the XBRL format). It is an open source platform available for all the common operating systems. It does not focus on content analysis of the XBRL format filings, it rather focuses on well-formedness checks and semantic validation given numerous XBRL-related standards. While *Arelle* is a powerful environment to process and search XBRL documents, to our knowledge it has no capabilities to include advanced NLP technologies for content analysis and graph databases for knowledge graph representations of content.

A more detailed description of the available NLP components, pipelines and technologies is presented in the following section, where we discuss the common technologies and their application in our specific scenario.

The business literature has demonstrated interest both in director networks and in the risks identified by firms, often relying on proprietary, curated databases focused on large firms. However, to the best of our knowledge, we are not aware of any publication that employs techniques such as our to accomplish these analytical goals at scale, i.e. a time mapping of relations between individuals and institutions, and the topic modeling analysis of identified risks and risk management strategies mapped onto the time axis over SEC filings. We welcome further feedback on additional research questions that this project's goals and methods may be well suited to address.

3. Implementation

Each of the SEC filings is composed of multiple sections, each of which may provide information of different scholarly interest. Thus, we employ Machine Learning (ML) (Scikit Learn (Buitinck et al., 2013; Pedregosa et al., 2011) and our own set of ML classifiers, e.g. using Bayesian or Support Vector Machine approaches). The models are extracted and the algorithms are trained on the manually annotated corpus that represents the distinct portions of SEC filings, in order to split them into appropriate segments for further analysis. For instance, a typical table of contents for the 10-K would include the following disclosures:

Item 1. Business (Description of the filing company's background, products, strategy and competition), Item 1A. Risk Factors (the factors that could most substantially affect the company's profitability and therefore the risk that shareholders may lose their investment), Item 3 Legal Proceedings, and so on. Additional key items of note include Management's Discussion and Analysis of Financial Condition and Results of Operations (Item 7) and Financial Statements (Item 8). Some items required to be part of a particular form may be incorporated by reference to another form. For instance, certain items in the 10-K are not provided therein, but are instead met by referencing the company's proxy statement.

Given the various sections and the unique content within each, we first pursued steps to delineate the content into sub-corpus that represented these component sections. We discuss these steps now: Creating the Training Corpus, De-

[1] For example, *Intelligize* (`www.intelligize.com`), a LexisNexis company, offers services related to the SEC EDGAR data set, that we have not analyzed in detail.

tection of Content in Paragraphs, and Natural Language Processing.

3.1. Creating the Training Corpus

We used the XML Path Language (XPath) (DeRose and Clark, 1999) to markup an initial training corpus from existing SEC reports manually using only a web-browser and a database table for the selected paragraphs and document reference. Our assumption is that the SEC filing URLs do not change, neither the final document structure and content for every single URL. Thus a corpus definition using XPath and URL tuples seems appropriate.

A team of corpus annotators browsed a selection of the SEC EDGAR reports using a web-browser that provided XPath information for selected paragraphs (e.g. Google Chrome). We collected those XPath descriptions with the document URLs in a data table for the target chapter types. The annotators marked the paragraphs of interest, in one particular case for example the sections of the 20-F which provided biographical information on one or more senior executives or board members of the company. This manual markup provided an XPath specification and a document URL referencing the start and end points of each sentence or paragraph within the document that was in-scope for the specified purpose.

The selected reference pointers to the remote documents were stored in database tables and validated by the corpus annotators.

We developed a crawler to fetch the text portions from the specific documents given the XPath description. The crawler aggregates the text paragraphs and generates a raw text corpus that serves as the training corpus for our machine learning (ML) content detection algorithms. This training corpus was used at the same time as one part of the textual data for extraction of information for the network study of individuals and firms, as well as the risk analysis using a topic modeling framework.

With this configuration we are able to process large volumes of corporate filings and extract the paragraphs of interest.

3.2. Detection of Content in Paragraphs

To train an algorithm to detect the desired content automatically, we used supervised machine-learning approaches. We utilized a Bayesian text similarity algorithm using frequency profiles of unigrams over the target text portions, as well as Support Vector Machines to detect the optimal hyperplane between binary models, e.g. CV vs. non-CV sections or risk-management vs. non-risk-management descriptions.

We assessed the performance of the algorithms on the training texts against the human-classification of the same texts. By automatic and manual modification of model weights assigned to the classes given the N-gram models, we improved the accuracy of the classifiers to predict either "relevant" or "not-relevant."

In basic evaluations we concluded that the ML-based algorithm can achieve a detection rate for sections with the curriculum vitae of the management or the description of board members at higher than 95%. As for the detection of sections discussing risk factors we achieve an accuracy above 90%.

Automatically detected relevant paragraphs that are not part of the manually generated and validated training corpus, are extracted and saved as a separate corpus for further analysis, including the relevant meta-information as for example the URL and XPath information.

3.3. Natural Language Processing

Next, our project requires the ability to parse the relevant text to extract focal information. We found that available parsers were inconsistent in their treatment of domain, time, and company specific information. Further, our use-case has considerable nuance in regard to pronoun-resolution and each individual may have time-variant characteristics or positions.

To evaluate the quality of the parsing and to ensure we obtained the highest-quality extraction of information, we built an interface to allow comparison of multiple parsing tools, including Stanford CoreNLP (Manning et al., 2014), spaCy,[2] NLTK (Bird et al., 2009) components, and various other NLP technologies.

The different NLP components provide linguistic analytical output for various domains in different quality. The majority of those tools are able to generate Part-of-Speech (PoS) tags for the text tokens (i.e. the words and punctuation marks). The problem is that the PoS-taggers make use of different tagsets, and the tagsets are utilizing a limited number of tags that ignore detailed morpho-syntactic features of high importance for a deep linguistic analysis. CoreNLP and spaCy segment text into sentences and generate for example dependency parse trees[3] for each sentence. These Dependency Trees describe rather shallow relations at the functional level of sentences (the subject and the object of the main predicate, etc.), leaving out essential information about scope and semantic hierarchies of sentential or clausal elements. A segmentation of clauses is not provided, but we were able to translate the Dependency Parse Trees and Constituent Trees that were generated by phrase structure parsers into clause segmentation information. Only CoreNLP provides an essential component for the resolution of anaphora and coreference (Clark and Manning, 2015; Clark and Manning, 2016).

To improve the named entity recognition and to add to the NLP components a sub-classification, we created an extended English morphological analyzer. We collected freely available resources with names of people known in the business world with their titles and roles. Additionally, we extended the analyzer with all variations of the name used in the business documentation. For example, Timothy Donald Cook, the CEO of Apple Inc., is also identified as the same entity referenced by strings like *Timothy* or *Tim Cook*. The morphological analyzer not only generates a named entity label for a **person name**, but also a sub-class ID for the **business domain**, and a unique identifier for the entity. The morphological analyzer includes also a list of

[2] The parser used in spaCy is an implementation of Honnibal and Johnson (2015).

[3] See for example (2014) for the CoreNLP Dependency Parser output.

to us known institutions and firm names with variants, e.g. *Apple Inc.* and simply *Apple*. The domains of institutions are emitted as tags as well, i.e. subtypes like **financial**, **IT**, or business types like **manufacturing** or **service** business. The morphological analyzer is implementation as a bidirectional Finite-state Transducer using the *Foma* (Hulden, 2009) compiler, with the named-entity databases containing extensive lists of people and institutions and detailed type information translated to *Lexc*[4] grammars.

Our goal was to utilize freely available NLP pipelines and components as much as possible, developing parallel processing chains that then validate the output via comparison. We developed uniform wrappers for CoreNLP, spaCy, OpenNLP, some NLTK components, and our own modules, that translate the NLP analyses into a uniform data structure (a Python class instance). The *LingData* class serves as this kind of wrapper. It translates PoS-tags into feature structures represented as Directed Acyclic Graphs. Constituent trees are translated into clause hierarchies and all scope relations between tokens and phrases are mapped to an API, allowing for requests like "does the main clause contain a sentential negation," or "is the embedded clause in the scope of future tense (from the matrix clause)." Such requests are wrapped into method or function calls within the *LingData* class. They are essential for advanced mapping of content at deep linguistic levels to Knowledge Graphs or Representations (using free graph representations or OWL-based ontologies). Dependency Graphs are similarly translated into data structures that can be queried for clause level dependencies, e.g. checking for core relations like *subject* and *object* of a *predicate*, mapping semantic triples from dependency representations. They are also mapped on phrase level dependencies, usually not obvious relations in a dependency parse tree, such that the entire *subject phrase* can be asked for. As long as our NLP components are able to resolve anaphoric expressions, the references for every pronoun and coreferent concepts are made available via methods in *LingData*. Entities and basic concepts in the NLP output are annotated for their synonyms, hypo-, and hypernyms using WordNet (Miller, 1995; Fellbaum, 1998) in NLTK.

Integrating the outputs of NLP components into one uniform data structure allows us to unify the outputs and identify mismatches, generate selection models, and weight those outputs to identify the best representation from often failing or insufficiently specific NLP results.

The mentioned NLP components exhibit systematic errors with specific constructions and sentence types. Common errors occur with constructions containing coordination, complex embeddings, constructions with gapping or ellipsis, and simply with very long sentences.[5]

To overcome issues and limitations with these purely data driven and probabilistic methods, we utilize hybrid approaches as in the Free Linguistic Environment (FLE) project (Cavar et al., 2016). Such approaches allow us to combine grammar engineering with data driven machine learning approaches to improve the NLP performance and generate deep linguistic annotations that go beyond the limited dependency parse or simple constituent structure trees. Our Probabilistic Lexical-functional Grammar approach in FLE combines and links hierarchical structural relations with functional annotations, elementary semantic and morpho-syntactic relations and properties. A detailed description of the deep linguistic analysis/annotation that can be achieved with a parser like FLE (or simpler forms implemented using NLTK components) would go beyond the scope of this abstract. We would be happy to discuss these properties in the full paper version and in the presentation of this work.

Our NLP pipelines are implemented as parallel Remote Procedure Call (RPC) (Microsystems, 1988) services that can be distributed over multiple instances and thus scaled for big data annotation. We provide wrappers in Python (and Go) for the mentioned pipelines. The code is mostly independent of the underlying operating system.

3.3.1. Topic Modeling of Risk Discussions

For topic modeling and analysis of the risk management content we used initially Mallet (McCallum, 2002). The large number of different implementations of topic modeling libraries in different systems allows us to extend our evaluations in various ways. Topic models generated by Mallet provide essentially a set of n groups of tokens from the text that represents n underlying or latent topics. Since the particular topic modeling technologies at the time of our experiments did not provide any systematic strategy to display variation of topics over time, we decided to generate the models for individual reports and generate a mapping of the topic related token list over the time axis by iterating the analyses.

Setting the number of topics to an initial 100, we used Mallet incrementing the optimization intervals that lead to an increasing number of empty topics, which we were inclined to take to be a good indicator for the optimal number of topics in the content of a particular report.

Our goal was to map these results on the different business domains and sectors, to identify time-dependent changes in the perception of risks and emerging mitigation strategies. This is ongoing research that we will report on independently.

3.4. Mapping Concepts and Relations on KRs

Our uniform *LingData* data structures contain detailed information about the structure of every single clause in the analyzed text. We are able to extract sentence internal clause structure, identify clause features and scope relations between clauses, core semantic relations within a clause (e.g. subject – predicate – object), as well as detailed properties of concepts, their semantic and morpho-syntactic features. This information is essential for the mapping of mentioned concepts on KRs.

Our graph-mapping approach includes the extraction of *subject – predicate – object* tuples from raw text. We validate the semantic relations using linguistic analyses, e.g. whether the utterance is embedded in a subjunctive or fu-

[4] See for details on the *Lexc*-formalism the documentation of *Foma* (Hulden, 2009).

[5] We are preparing an independent documentation of the fallacies of the common NLP pipelines and tools, and systematic issues with the inherent probabilistic technologies used in those.

ture tense context, or whether it is in the scope of such a context, or negative operators. The relations that are identified as valid facts and assertive statements are added to the graph, representing the subject and object as concepts, and the predicate as a relation with a time-stamp and meta-information about the source.

As an example, we can differentiate future tense claims like *Facebook will buy Oculus* from past tense assertions like *Facebook bought Oculus*. The latter allows us to extend the KR with a factual assertion of concepts and relations *Facebook* → *buy* → *Oculus*, while the future tense clause does not justify this mapping. Analyzing the same factual relation in an embedded clause as in *I would not claim that Facebook bought Oculus* in the scope of a negated matrix clause cannot be rendered as a factual assertion.

There are multiple interesting aspects to emphasize here, when it comes to the correct mapping of semantic relations extracted from NLP outputs. Due to space limitations we confine ourselves to this particular example.

As a graph database we utilize commercial products, Neo4J (`neo4j.com`) and Stardog (`www.stardog.com`). While both graph DBs offer similar capabilities with respect to graph storage and retrieval functionalities, Stardog provides a standardized SPARQL (The W3C SPARQL Working Group, 2013) interface and it offers the possibility to integrate Ontologies in form of OWL (W3C OWL Working Group, 2012; W3C OWL Working Group, 2009) and utilize the integrated Pellet reasoner (Sirin et al., 2007). Using ontologies in the graph DB allows us to infer a wider array of information about a concept by inheriting properties from the general class definition. As an example, if we assert that *Tim Cook isA CEO*, and if our ontology encodes the fact that *CEO isA Human*, we can infer that *Tim Cook* must be a human with all the attributes and relations that follow from that, without this being explicitly mentioned in any text. At the same time, an assertion like *Apple isA Institution* cannot be followed by an obviously wrong assertion like *Apple isA CEO (of Google)*, if the concept *Institution* is not subsumed under the concept *Human* in the ontology.

4. Discussion

Corporate reporting with the Securities and Exchange Commission (SEC) represents a significant, recurring domain-specific corpus with tremendous academic and practical value. To date, most research on corporate filings has employed limited parsing techniques or only sentiment or dictionary-based approaches. Our work applies the latest advances in linguistic approaches to one of the largest, publicly available corpus of business documents. In doing so, we make significant contributions to advancing the techniques of deep natural language processing. In particular, linguists currently seek to determine how to make their tools more specifiable to particular domains as well as to account for rapid-changes in terminology use. Further, we bridge linguistic processing techniques, including mapping of semantic properties, with network analysis of individuals and entities.

In addition, we provide a valuable platform for future work in business domains. Accounting, finance and management scholars all rely heavily on SEC filings, including 10-Ks, 10Qs, and 20-Fs, and these scholars are increasingly seeking to employ various forms of computer aided text analysis. These include for instance the readability of the disclosures (Loughran and McDonald, 2014), dictionary-based approaches (Andriy Bodnaruk and McDonald, 2015) including sentiment analysis and Naive Bayesian approaches (Loughran and McDonald, 2016). Scholars in the business domain are seeking to move beyond single-word approaches into phrase (n-gram) approaches that also draw on a greater wealth of linguistic components (Pandey and Pandey, 2017). Our pipeline extends even further, showing how deep semantic processing allows for the identification of previously unobservable relationships, by providing a mechanisms for surfacing deep interrelationships between concepts, entities and related parties. Our pipeline also allows for more traditional analysis of the word content in SEC disclosures.

Since we are lacking a gold standard corpus or data set for most of our tasks, we are not yet able to measure the effectiveness of extracting concepts and relations in a formal and quantitative way. The only evaluation criterion that we can apply is whether by human judgment the results are true and useful. The same can be applied to topic modeling results from the risk description sections. We hope to be able to provide much better evaluation criteria in the near future, given that we are able to generate corpora and initial data sets using our architecture.

5. Acknowledgments

We are grateful for the programming expertise of Lwin Moe and other valuable contributions from Katherine Spoon, Jonathan Anderson, Prateek Srivastava, Anthony Meyer, Ji Li, Celine Marquet, Atreyee Mukherjee and students in Dr. Cavar's Fall 2017 AI and Machine Learning seminar at Indiana University. We received useful comments and suggestions related to this research from colleagues at Indiana University, particularly participants in our workshop at the IU Network Institute. Cavar and Josefy are Fellows of the National Center for the Middle Market at the Fisher College of Business, The Ohio State University, and acknowledge the Center's support for this research, which enabled us to gather and validate data on middle market firms for inclusion in the larger dataset. This research is also funded by the Office of the Provost of Research at Indiana University in form of a collaborative interdisciplinary research grant.

6. Bibliographical References

Andriy Bodnaruk, T. L. and McDonald, B. (2015). Using 10-k text to gauge financial constraints. *Journal of Financial and Quantitative Analysis*, 50(4):623–646.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

Blei, D. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Cavar, D., Moe, L., Hu, H., and Steimel, K. (2016). Preliminary results from the free linguistic environment project. In Doug Arnold, et al., editors, *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, pages 161–181. CSLI Publications.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP 2014*.

Clark, K. and Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*.

Clark, K. and Manning, C. D. (2016). Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing*.

DeRose, S. and Clark, J. (1999). XML path language (XPath) version 1.0. W3C recommendation, W3C, November. http://www.w3.org/TR/1999/REC-xpath-19991116/.

Engel, P., Hamscher, W., Shuetrim, G., vun Kannon, D., and Wallis, H. (2013). Extensible business reporting language (xbrl) version 2.1. XBLR recommendation, XBRL International, February. http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Fischer, H. and Mueller, D. (2011a). Enabling comparability and data mining with the Arelle open source unified model, September. Paper presented at First Conference on Financial Reporting in the 21st Century: Standards, Technology, and Tools.

Fischer, H. and Mueller, D. (2011b). Open source & XBRL: the Arelle project, April. Paper presented at 5th University of Kansas Conference on XBRL.

Fischer, H. (2013). Evolution and future trends for XBRL development, April. Paper presented at 6th University of Kansas Conference on XBRL.

Honnibal, M. and Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September. Association for Computational Linguistics.

Hulden, M. (2009). Foma: A finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32. Association for Computational Linguistics.

Jonathan L. Rogers, D. J. S. and Zechman, S. L. (2017). Run edgar run: Sec dissemination in a high-frequency world. *Journal of Accounting Research*, 55(2):459–505.

Josefy, M., Harrison, J., Sirmon, D., and Carnes, C. (2017). Living and dying: Synthesizing the literature on firm survival and failure across stages of development. *Academy of Management Annals*, 11(2):770–799.

Loughran, T. and McDonald, B. (2014). Measuring readability in financial disclosures. *Journal of Finance*, 69(4):1643–1671.

Loughran, T. and McDonald, B. (2016). Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, 54(4):1187–1230.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

McCallum, A. K. (2002). MALLET: A machine learning for language toolkit.

Microsystems, S. (1988). RPC: Remote procedure call protocol specification. Request For Comment (RFC) 1057.

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Newman, M., Barabási, A.-L., and Watts, D. J. (2006). *The Structure and Dynamics of Networks*. Princeton Studies in Complexity. Princeton University Press.

Pandey, S. and Pandey, S. K. (2017). Applying natural language processing capabilities in computerized textual analysis to measure organizational culture. *Organizational Research Methods*, In-press:1–33.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53. Software Engineering and the Semantic Web.

The W3C SPARQL Working Group. (2013). SPARQL 1.1 overview. W3C recommendation, W3C, March. http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/.

W3C OWL Working Group. (2009). OWL 2 web ontology language document overview. W3C recommendation, W3C, October. http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.

W3C OWL Working Group. (2012). OWL 2 web ontology language document overview (second edition). W3C recommendation, W3C, December. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.