Web Text-based Network Industry Classifications: Preliminary Results

Eric Heiden
University of Southern California
heiden@usc.edu

Gerard Hoberg
University of Southern California
hoberg@marshall.usc.edu

Craig A. Knoblock
University of Southern California
knoblock@isi.edu

Palak Modi
University of Southern California
palakmod@usc.edu

Gordon Phillips
Dartmouth College
Gordon.M.Phillips@dartmouth.edu

Gaurangi Raul
University of Southern California
gaurangr@usc.edu

Pedro Szekely
University of Southern California
pszekely@isi.edu

ABSTRACT

Studies of market structure and product market competition are important in many disciplines, such as economics, finance, accounting and management. Reliable data for such studies is easily available for public firms (e.g., 10-K filings), but no reliable data exists for private firms. In this work we propose to mine the Internet Archive Wayback Machine, a digital archive of the World Wide Web, to build a database of 300,000 companies to support analyses of market structure, product market competition, and innovation. The goal of the WTNIC project is to download pages from the archive to build a profile for each company, and to use machine learning techniques to define similarity between companies based on similarity of their product and service offerings. This paper describes the challenges that must be overcome, our approach to overcome these challenges, and some preliminary results.

KEYWORDS

Document classification, company similarity, competitive landscape, industry classifications, TNIC

ACM Reference format:
Eric Heiden, Gerard Hoberg, Craig A. Knoblock, Palak Modi, Gordon Phillips, Gaurangi Raul, and Pedro Szekely. 2017. Web Text-based Network Industry Classifications: Preliminary Results. In *Proceedings of DSMM'17, Chicago, IL, USA, May 14, 2017,* 3 pages.
DOI: http://dx.doi.org/10.1145/3077240.3077245

1 INTRODUCTION

In prior work, Hoberg and Phillips [4] built TNIC, a database that measures company similarity based on similarity of their product and services. Hoberg and Phillips built the database for publicly traded companies using the business description sections from 10-K annual filings, which companies are legally required to file every year. The database, containing data for about 5,000 companies since 1996, has become a widely used resource for a large number of analyses by a large number of researchers.

The goal of our new work is to significantly expand the coverage of the database, from 5,000 publicly traded companies to about 300,000 companies, including private companies. Because there is no reliable source of data for private companies that is analogous to the 10-K filings for public companies, we use company web sites as a source of data. To support historical analyses, we propose to acquire the pages from the Wayback Machine, a digital archive of the World Wide Web that contains a cache of the web since 1996, with over 450 billion pages.

Our approach is to use the web pages of a company to create a feature vector for each company, representing the products and services it offers. The database will contain the vectors for 300,000 companies and will be used in a variety of machine learning algorithms to identify clusters of competitors and evolution of industries over time. In the rest of this paper we present challenges and approaches, preliminary results, and close with conclusions and directions for future work.

2 APPROACH

We formalize the problem of computing a company similarity database as a tuple $< C, \mathcal{P}, \mathcal{Y}, pages, O, sim >$ where C is a set of companies, $\mathcal{P}$ is the set of web pages in the Wayback Machine, $\mathcal{Y}$ is the set of years for which the Wayback Machine has pages, $pages(c, y)$ defines the set of pages $p \in \mathcal{P}$ published in the company's web site in year $y \in \mathcal{Y}$, O is an oracle that defines company similarity, and *sim* defines company similarity in terms of web pages.

As it is difficult to assemble a comprehensive ground truth of company competitors, we define an easy-to-build oracle to support formal evaluation. The oracle O is a tuple $< C^f, C^o, competitors >$ where $C^f \subseteq C$ is a set of focus companies for which the oracle has competitor answers, $C^o \subseteq C$ is the set of companies that are considered as competitors of the focus companies, $competitors(c, y) \subseteq C^o$ are the competitors of a company $c \in C^f$ in year $y \in \mathcal{Y}$.

The objective of our work is to compute company similarity using the web pages published on the Wayback Machine. Our goal

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DSMM'17, Chicago, IL, USA

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.

978-1-4503-5031-0/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3077240.3077245

is to construct a *sim* function such that for all $x \in competitors(c)$ and all $y \notin competitors(c)$

$$sim(pages^*(c), pages^*(x)) \geq sim(pages^*(c), pages^*(y))$$

where $pages^* \subseteq pages$ are the company pages downloaded for processing.

Given these definitions, a proposed function sim_0 can be evaluated with an oracle O using normalized discounted cumulative gain (NDCG) on a ranking produced using *sim*.

Naive Implementation (sim_n): of similarity, would use cosine similarity on vectors built using the words in $pages(x), \forall x \in C^o$. Such a definition is impractical as the number of pages for large companies is very large, and too time-consuming and expensive to download. Furthermore, in addition to products and services, company web sites include information on many topics, such as customer support, marketing, employment, legal issues, etc. Including the words from all pages leads to poor results because similarities on non-product and service pages may overwhelm similarities of product and services. Our implementation of sim_n using a sample of approximately 12,000 pages from 34 companies produced worse than random results for all 5 focus companies in our oracle (e.g., for 3 companies, all records with non-zero cosine similarity were incorrect).

Breadth-First Limited Crawling ($breadth\_first(limit, depth)$): this definition of the $pages^*$ function navigates company web sites breath-first, starting at the company home page, visiting pages at most *depth* links away from the home page, downloading at most *limit* number of pages.

Word Glossaries. Our experiments with sim_n revealed that it is important to select words that characterize products and services. Unfortunately, there are no available product and services glossaries providing comprehensive coverage on all industry types.

We thus consider two resources: text extracted from 10-K business descriptions, and text from the 2017 NAICS manual produced by the Census Bureau. The 10-K corpus contains business descriptions (each roughly 2-10 pages) from over 4,000 filers in 2015. The NAICS manual contains 517 pages describing the products sold by firms in various industries.

We developed an automated approach to mine product and service glossaries from the 10-K filing. The algorithm defines a $10k(c)$ function that maps a company c to the first 600 words of the *description* section of the 10-K document. We only use the first 600 words of the description in order to get similar-length texts from all companies.

In a second step, we process the descriptions using a part-of-speech tagger [1] and use the tags to define three base glossaries: g^n containing nouns (20,614 words), g^p containing proper nouns (37,388 words) and g^a containing adjectives (13,218 words). We then construct 7 glossaries using a union of all combinations of the 3 base glossaries.

Many words in the glossaries produced in the previous step occur infrequently in the union of $10k(c)$ for all $c \in C^o$. To eliminate such words, our algorithm computes the number of times a word is tagged as noun, $count(w, noun)$ for which $w \in 10k(c)$ and then removes all words from the noun glossary for which $count(w, noun)/count(w) < T$. We have done similar screening for the glossaries containing adjectives and proper nouns. We achieved the best results for $T = 0.1$.

Some words occur frequently across many companies, which makes them less useful in identifying the most closely related companies. In the final step our algorithm computes the number of companies $n(w)$ for which $w \in 10k(c)$, and then removes from the glossaries all words w for which $n(w)/|C^o| > K$. We set $K = 0.2$ based on our previous experiments in the TNIC project.

Document Processing. The inputs to this process are $pages^*(c)$ and a glossary g (one of the 7 glossaries mentioned above). A data processing pipeline processes the pages in several steps: a data cleaning step extracts all text from the pages, removing HTML tags and numbers, and combines the text into a single document; a tokenization step produces a word list; and a glossary extraction step intersects the word list with the glossary g. The output is a list of $(word, count)$ pairs for each $c \in C^o$, recording the number of times each $word \in g$ occurs in $pages^*(c)$.

Similarity Function. Our similarity function $sim^{g(K),m}$ constructs a feature vector for each company c using the $(word, count)$ pairs computed in the document processing step. m can be any metric suitable for comparing feature vectors, such as cosine or Jaccard similarity.

3 EVALUATION

To evaluate our approach, we defined an oracle O using 5 focus companies in different industries. For each focus company c, our business school experts manually defined $competitors(c)$ as required by our oracle definition. The resulting C^o contains 34 companies. The list below shows the focus companies and their close competitors (the codes in parenthesis are identifiers to refer to the companies in the text below).

- 1013 Communications (C1): Alternative Newsmedia (C28) MediaDC (C29) BH Courier (C30) Dallas Observer (C31) Boulder Weekly (C32) Miami New Times (C33)
- Ajubeo (C2): Enstratius (C6) Cumo Logic (C7) CTL IO (C8) Egnyte (C9) Apprenda (C10) EMC (C11) Gigaspaces (C12) Amazon Web Services (C13) Google Cloud Platform (C14) Cloudera (C15)
- Anderson Burton Constructions (C3): Holder Construction (C16) ENR (C17) Kiewit (C18)
- BK Creative (C4): Studio SFP (C19) Red Shine Studios (C20) CMY Ken (C21) 99 Designs (C22) jSnyderDesign (C23)
- Black Abbey Brewing (C5): Samuel Adams (C24) New Belgium Brewing (C25) Midnight Brewery (C26) Bellavance Beverage Company (C27)

We evaluated our $sim^{g(K),m}$ algorithm using the 7 glossary combinations described above, using $K = 0.2$, $T = 0.1$ and $T = 0.2$, and using $m \in \{jaccard, tfidf\}$. The best results, shown in Table 1 were obtained with the glossary containing nouns, proper nouns and adjectives, $K = 0.2$, $T = 0.1$, and $m = tfidf$. Each row lists the focus company in the first column, the NDCG score and the rank of each of the competitors; the results in boldface indicate incorrect results.

These results are a significant improvement over the naive baseline sim_n. The new algorithm found the competitors for all 5 focus companies and had only a few errors, shown in boldface in Table 1. The results for C3 are suboptimal because $pages{*}(C3)$ contains only

	NDCG	Focus company(rank)
C1	0.94	C28(1) C29(2) **C16(3)** C30(4) C31(5) C32(6) **C21(7)** **C6(8)** C33(9)
C2	1.00	C6(1) C7(2) C8(3) C9(4) C10(5) C11(6) C12(7) C13(8) C14(9) C15 (10)
C3	0.97	C16(1) C17(2) **C4(3)** C18(4)
C4	1.00	C19(1) C20(2) C21(3) C22(4) C23(5)
C5	1.00	C24(1) C25(2) C26(3) C27(4)

Table 1: Evaluation results

one page, a problem that we are currently investigating. The suboptimal result for C1 illustrates that some industry types may prove to be difficult to classify. C1 is an media/news company, and the majority of $pages*(C5)$ are content pages (stories, articles, photographs, design work, etc.) rather than descriptions of the products and services.

4 CONCLUSION AND FUTURE WORK

Our initial work indicates that web text-based industry classifications are not only feasible to create, but they also offer excellent potential for impacting research exploring the market structure of private versus public firms, market structure, product market competition, and successful innovation of earlier-stage firms.

We have a number of directions planned for future work. First, we plan to explore different scoring algorithms besides TF-IDF with cosine similarity, such as Jaccard Similarity [2], which was used in TNIC, and Okapi Best Matching 25 (BM25) [5], which yielded the best results in our previous tests in Elasticsearch [3]. Second, we plan to do a comparison on the original set of 4,000 companies that were analyzed in TNIC [4] to see how our method of using web pages compares with the original method of using the 10-K reports. Third, we plan to collect data for many more companies and data for those companies spanning the last 20 years so that we can analyze how the competitive landscape changes over time in various markets. Finally, we plan on making an analytical web-based tool for researchers to access the resulting data using ElasticSearch. This will allow users to query our database of public and private firms and identify their competitors, accompanied with a pairwise similarity scores indicating the strength of competitive links.

ACKNOWLEDGEMENTS

This material is based on research sponsored in part by the National Science Foundation under Grant Nos. 1561057 and 1561068.

REFERENCES

[1] Eric Brill. 2000. Part-of-speech tagging. In *Handbook of natural language processing*. CRC Press.

[2] AnHai Doan, Alon Halevy, and Zachary Ives. 2012. *Principles of data integration.* Elsevier.

[3] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide.* ” O’Reilly Media, Inc.”.

[4] Gerard Hoberg and Gordon Phillips. 2016. Text-based network industries and endogenous product differentiation. *Journal of Political Economy* 124, 5 (2016), 1423–1465.

[5] Stephen Robertson, Hugo Zaragoza, and others. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.