CompeGe: Computing Company Competitor Pairs By Knowledge Based Inference Combined With Empirical Validation

Georg Gottlob[1], Jinsong Guo[1,*,†], Aditya Jami[2], Markus Kröll[2], Stéphane Reissfelder[2], Lukas Schweizer[2], Eric Aichinger[2] and Stefano Sferrazza[2]

[1]*Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK*

[2]*Meltwater, San Francisco, California, USA*

Abstract

This paper is an interim report about an industrial application that uses Datalog combined with empirical methods to compute competitor information from a knowledge graph. The Owler knowledge graph is one of the world's largest companies information systems (CIS). It contains data about 16+ million companies crowd-sourced from over 1 million experts. In particular, for most companies, it contains a set of competitors. Such competitor information is very useful for many B2B applications such as *lead generation*. However, competitor relations in crowd-sourced CIS are naturally incomplete. This paper presents CompeGen, a method that applies Vadalog (a particular variant of Datalog) rules to infer new competitor pairs from existing ones in the Owler CIS. Since using the Vadalog inference program alone is insufficient, CompeGen combines its inference process with a "learning" process to acquire some required logical facts and further validates the inference results via an empirical validation process. CompeGen was tested using the companies belonging to the "Internet Software" sector in Owler. It has discovered 23,180 new competitors of which over 80% were correct. We are improving the system and will report further results in the full paper.

Keywords

Datalog applications, Vadalog, competitor computation, knowledge-based inference, empirical validation

Datalog 2.0 2022: 4th International Workshop on the Resurgence of Datalog in Academia and Industry, September 05, 2022, Genova - Nervi, Italy

*Corresponding author.

†This work was done while Jinsong Guo was working as an external consultant for DeepReason.ai.

georg.gottlob@cs.ox.ac.uk (G. Gottlob); jinsong.guo@cs.ox.ac.uk (J. Guo); aditya.jami@meltwater.com (A. Jami); markus.kroell@meltwater.com (M. Kröll); stephane.reissfelder@meltwater.com (S. Reissfelder); lukas.schweizer@meltwater.com (L. Schweizer); eric.aichinger@meltwater.com (E. Aichinger); stefano.sferrazza@meltwater.com (S. Sferrazza)

0000-0002-2353-5230 (G. Gottlob); 0000-0002-1142-3610 (J. Guo); 0000-0003-1167-1777 (L. Schweizer); 0000-0001-9569-4428 (S. Sferrazza)

1. Introduction

In today's global competitive environment, competitor data constitutes information useful to many business applications, such as *Competitive Intelligence*, *Lead Generation*, *Recommender Systems*, and so on. For example, for lead generation, the competitors of a customer A of some company C may be good sales leads for C, as they are likely to require the same products or services as A. Competition-data sellers usually maintain a manually curated database or knowledge graph containing competitor pairs as part of a companies information system (CIS) that also maintains other useful information about companies such as the industry sectors in which they operate. The Owler[1] knowledge graph (a.k.a., the Owler competitive graph, see the graph on the left of Fig. 1 as an example) is one of the world's largest CIS. It contains data about 16+ million companies crowd-sourced from over 1 million experts. However, competitor relations in such crowd-sourced CIS are naturally rather incomplete. The **main goal** that we target is to infer new competitor pairs from existing competitor pairs in the Owler CIS.

Datalog programs that perform inference based on knowledge about companies are naturally suitable for inferring new competitor pairs from existing ones in a CIS. However, using a Datalog program alone is not sufficient because the required knowledge may be absent, and the inference results are sometimes inaccurate. In this paper, we present CompeGen designed based on our **key idea**:

Key Idea: Combine a knowledge-based inference process with an upstream knowledge-learning process and a downstream validation process.

CompeGen "learns" new knowledge from the existing knowledge in Owler, and then performs an inference process based on both the learned and the existing knowledge about companies. The inference results are further validated by an empirical validation process based on co-occurrences of companies in documents of a document repository. The combination of (i) knowledge about companies [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], and (ii) co-occurrences of companies in documents of a document repository [11, 12, 13, 14, 15], distinguishes CompeGen from conventional methods that leverage either one or the other of the two tools without combining them. Fig. 1 illustrates the workflow of CompeGen, which will be detailed in the next two sections.

2. Knowledge-based Inference

The inference process of CompeGen uses several different types of knowledge (represented as logical facts) about companies: (a) Competitor(A, B, s) which represents an existing competitor pair in Owler that a company A has a competitor of B with a proximity score s ranging from 0 (extremely unlikely to compete) to 100000 (sure competitors); (b) CompSector(C1,S1) which represents the fact that a company identified by the company ID C1 belongs to the industry sector S1; (c) CompatSec(S1,S2,c) which represents the fact that two sectors S1 and S2 are compatible sectors with a sector compatibility score of c. Two sectors, S and S', are compatible if there are more than a certain amount of competitor pairs in each of which one company belongs

[1] https://corp.owler.com/

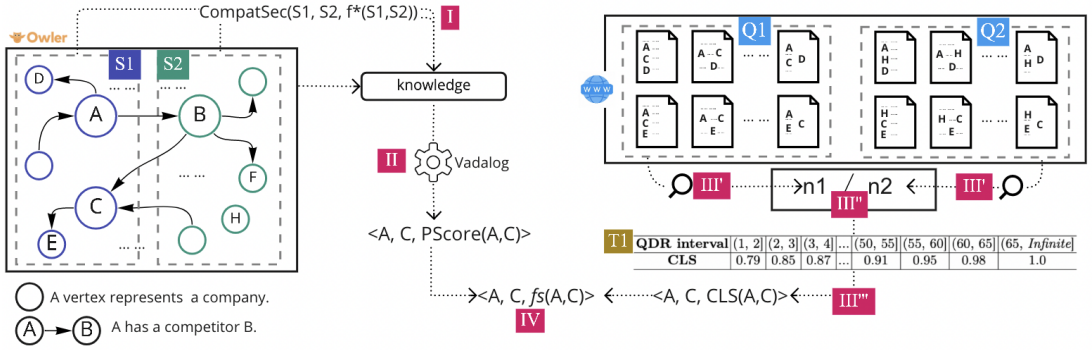

Figure 1: A simple example of the workflow of CompeGen.

to S and the other belongs to S' (see Section 3). For example, in Figure 1, the sectors S1 and S2 are compatible. Knowledge of types (a) and (b) comes directly from the Owler CIS while the knowledge about sector compatibilities is learned from existing knowledge in the Owler CIS, which is explained in the next section. Based on such knowledge, CompeGen computes candidate competitor pairs via Vadalog, which is a particular variant of Datalog well-suited for knowledge graphs [16]:

```
Cand(C1,C2,PScore):-Competitor(C1,C2,PScore).
Cand(C1,C3,PScore):-Cand(C1,C2,PS12),Competitor(C2,C3,PS23),
    CompSector(C1,SEC1),CompSector(C3,SEC3),C1!=C3,
    CompatSec(SEC1,SEC3,SeCoScore),Penalty(A), Cutoff(B),
    PScore=max((PS12+PS23-100000-A)*SeCoScore), PScore>B.
```

Each candidate competitor pair (C1,C3) computed by the above program is represented by the fact Cand(C1,C3,PScore) where PScore is a plausibility score expressing a degree of plausibility that company C3 is a competitor of company C1. The above Vadalog program computes a new fact Cand(C1,C3,PScore), either if such a fact is already in the Competitor relation, or if there is an already computed fact Cand(C1,C2,PS12) and a fact Competitor(C2,C3,PS23), where the certain conditions are satisfied. These conditions require that C1 be different from C3, and that the computed plausibility score PScore be larger than some cutoff constant B (represented as Cutoff(B)), where PScore is the maximum value of $(\text{PS12} + \text{PS23} - 100000 - A) \times SeCoScore$ (SeCoScore is the compatibility score of the sectors of C1 and of C3) over all matching choices of C2, SEC1, and SEC3. A penalty constant A (represented as Penalty(A)), with $0 < A < 100000$, lowers the proximity scores of new candidate pairs generated by transitivity.

3. System Overview

The CompeGen approach can be intuitively explained via the main steps described below.

First, at Step I, a knowledge-learning process is performed to learn the knowledge about sector compatibilities, i.e., sector compatibility scores, from the data in the Owler CIS. Let S be a sector and $S_1, \ldots, S_n$ be all sectors such that there is at least one competitor pair from S and S_i, i.e., there is one edge from S to S_i in the competitive graph, for $1 \leq i \leq n$. For every such S_i, N_i is defined to be the number of edges from S to S_i. Let $N = \max_{1 \leq i \leq n}(N_i)$. Let $c = 0.2$

be some empirically decided cutoff-constant. If $\frac{N_i}{N} < c$ this then means that S_i and S are not compatible, and thus the compatibility weight $f(S, S_i) = 0$. Otherwise, $f(S, S_i)$ is calculated via an empirically determined function $f(S, S_i) = 1 - (1 - \frac{N_i}{N})^m$, where $m = 3$. A smaller m, such as 1, may cause $f(S, S_i)$ to be lower than expected, especially when N is very large while N_i is also large but much smaller than N. For example, when $N = 20000$ and $N_i = 10000$, $f(S, S_i) = 0.5$ if $m = 1$, while $f(S, S_i) = 0.875$ if $m = 3$, and the latter is more reasonable. The compatibility score $f^*(S, S_i)$ is the maximum of the compatibility weights $f(S, S_i)$ and $f(S_i, S)$.

Next, at Step II, candidate competitor pairs, e.g., (A,C) in Fig. 1, are generated via the Vadalog program described in Section 2.

The next task (Step III) is to validate each generated candidate competitor pair, e.g., (A,C), against a document repository. The Web is used as the document repository in CompeGen. A Competition Likelihood Score CLS(A,C) of A and C, ranges from 0 (not competitors) to 1 (competitors), is determined based on the co-occurrences of A and C in different Web pages. CLS(A,C) is calculated based on a comparison of a number of search results for two groups of queries to the document repository (Step III'): (i) a first group of queries, for co-occurrences of names of A and of C together with names of some competitors A_i^* of A (if any), such as D, or some competitors C_i^* of C (if any), such as E. (ii) a second group of queries, corresponding to the first queries, where either A or C is replaced by random companies R(A) or R(C) from the CIS not known to be in a competitor relationship with A or C, such as H. Examples of web pages that match these two groups of queries are in dashed boxes labeled Q1 and Q2, respectively. From the average number of search results of queries in query groups (i) and (ii), denoted by n_1 and n_2, respectively, a Query-result Difference Ratio (QDR) is calculated by n_1/n_2 (Step III"). Based on the QDR, the likelihood score CLS(A,C) is calculated (Step III"') according to a predefined QDR-to-CLS lookup table, such as T1, whereby a higher CLS is achieved if the QDR is larger.

The final part (Step IV) computes for each candidate pair (A,C) a final proximity score via: $fs(A, C) = \frac{PScore(A,C)+CLS(A,C)\times 10^5}{2}$. If (i) $fs(A, C)$ is larger than a given constant (e.g., 90000), and (ii) (A,C) is not already stored in the Owler CIS with a score $s \geq fs(A, C)$ then (A,C) is inserted into the CIS with $fs(A, C)$.

4. Evaluation

CompeGen was tested using the companies belonging to the "Internet Software" sector in Owler. CompeGen has discovered 23,180 new competitor pairs. We randomly sampled 200 of these new competitor pairs and asked internal experts to validate these 200 pairs' correctness. It turned out that 162 pairs were correct, thus the precision of CompeGen is around 0.81. Since the total number of missing competitor pairs in Owler was unknown, it was hard to evaluate the recall of CompeGen. However, discovering 23,180 new competitor pairs with a precision of 0.81 for a single sector has shown the potential to improve the Owler CIS using CompeGen. That has led to Meltwater's adoption of CompeGen in the completion task of Owler. We are making several improvements to CompeGen, of which the results will be reported in the full paper.

References

[1] B. H. Clark, D. B. Montgomery, Managerial identification of competitors, Journal of Marketing 63 (1999) 67–83.

[2] J. F. Porac, H. Thomas, Taxonomic mental models in competitor definition, Academy of management Review 15 (1990) 224–240.

[3] T. Lappas, G. Valkanas, D. Gunopulos, Efficient and domain-invariant competitor mining, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 408–416.

[4] G. Valkanas, T. Lappas, D. Gunopulos, Mining competitors from large unstructured datasets, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 1971–1984.

[5] N. Joseph, B. S. Kumar, Top-k competitor trust mining and customer behavior investigation using data mining technique, Journal of Network Communications and Emerging Technologies (JNCET) 8 (2018) 26–30.

[6] G. Pant, O. R. Sheng, Web footprints of firms: Using online isomorphism for competitor identification, Information Systems Research 26 (2015) 188–209.

[7] T.-N. Doan, F. C. T. Chua, E.-P. Lim, Mining business competitiveness from user visitation data, in: International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, Springer, 2015, pp. 283–289.

[8] N. Raman, G. Bang, A. Nematzadeh, Multigraph attention network for analyzing company relations, in: Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition, 2019, pp. 426–433.

[9] Y.-P. Chen, T.-L. Hsu, W.-K. Chung, S.-C. Dai, L.-W. Ku, Upstream, downstream or competitor? detecting company relations for commercial activities, in: International Conference on Human-Computer Interaction, Springer, 2019, pp. 42–52.

[10] W. Qin, X. Luo, H. Wang, Implicit business competitor inference using heterogeneous knowledge graph, in: 2021 IEEE International Conference on Big Knowledge (ICBK), IEEE, 2021, pp. 198–205.

[11] R. Li, S. Bao, J. Wang, Y. Yu, Y. Cao, Cominer: An effective algorithm for mining competitors from the web, in: Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 948–952.

[12] R. Li, S. Bao, J. Wang, Y. Liu, Y. Yu, Web scale competitor discovery using mutual information, in: International Conference on Advanced Data Mining and Applications, Springer, 2006, pp. 798–808.

[13] S. Bao, R. Li, Y. Yu, Y. Cao, Competitor mining with the web, IEEE Transactions on Knowledge and Data Engineering 20 (2008) 1297–1310.

[14] R. L. Cilibrasi, P. M. Vitanyi, The google similarity distance, IEEE Transactions on knowledge and data engineering 19 (2007) 370–383.

[15] O. P. Damani, Improving pointwise mutual information (pmi) by incorporating significant co-occurrence, arXiv preprint arXiv:1307.0596 (2013).

[16] L. Bellomarini, E. Sallinger, G. Gottlob, The vadalog system: datalog-based reasoning for knowledge graphs, Proceedings of the VLDB Endowment 11 (2018) 975–987.