A Study on Object Search and Relationship Search from Text Archive Data

Yating Zhang

ABSTRACT

Large-scale text archive data are prevalent in many areas including informatics, computational social science and finance. However, to better explore and understand archival datasets, still many research problems remain. (1) The first problem lies in the difficulties of searching in the archives. Often searchers face the problem of the gap between the knowledge they possess and the content they desire to search due to domain differences or due to long time spans between documents contained in the archives. We call such problem a terminology gap. In such sense, it is sometimes difficult for the users to come up with a right key word to search the object in an unknown domain. (2) Besides the object search, searching for the relationships in archives is another challenging task. For example, searching for the set of evidences to explain the similarity relationship among entities. (3) In addition, the cause-and-effect relationship is another unknown relationship likely to be searched by the users who wish to discover and understand the changes reflected in text archives. For example, users may not be aware of the changes in one object causing the changes related to another object. This thesis aims at establishing methodologies for supporting object search as well as relationship search in text archive data. More specifically, we address the following three research topics:

1. **Detecting Semantically Similar Terms across Different Domains**
   We address the problem of terminology gap in object search by detecting semantically similar terms across different domains. For example, users searching for documents about music devices in 1980s which are similar to iPod, may fail to succeed since they may not know Walkman which played similar role as iPod does nowadays. We solve this problem by finding the counterpart (e.g., Walkman) of the query (e.g., iPod) that existed in the target domain (e.g., 1980s). We propose an efficient method to find counterparts by transforming the representation of terms within different vector spaces. We then enhance such global correspondence method by considering also the local context of terms (local correspondence). Through the extensive experiments conducted on the 20-years long archive, the New York Times Annotated Corpus, as well as the 200-years long archive, the Times

Archive, the proposed methods have been proved to be effective in searching for different types of entities (objects, persons and locations).

2. **Detecting Similarities between Entities from Different Domains**
   We introduce a novel problem of automatically explaining similarity of entities from different domains to let users understand and "explore" a given unknown domain through the comparison with the domain familiar to them. We propose an effective approach for this task, which, in a fully unsupervised fashion, returns a set of evidences indicating similar aspects of entities dispersed across different domains. In particular, we detect their commonalities as well as aligned differences by proposing two different approaches for selecting explanatory terms. We evaluate the proposed methods on the New York Times Archive and we demonstrate that they can successfully detect both commonalities and aligned differences of compared entities. To better present the results discovered in the above two topics, we introduce two views for result investigation: the first one, called the top counterpart view, visualizes the top temporal counterparts of a queried entity, while the second one, the similarity explanation view, displays the extracted evidences to support the understanding of across-domain similarity between the query and a selected counterpart.

3. **Detecting Cause-Effect Relationships in Text Archive**
   We investigate how the objects change and what are the effects of these changes by detecting causal relations in a specific type of text archive, online product reviews. We are particularly interested in understanding social impacts of technology and in discovering how changes of product features influence changes in our social lives. We address this problem by providing novel methodologies to extract evidences of technology impact on human activities. In particular, we first distinguish two kinds of product-related terms: terms denoting physical product features used for representing the potential causes and terms describing situations when products are used as the potential effects. We then detect changes in both the types of terms over time by tracking fluctuations in their popularity and usage. Finally, we discover cases when changes of product features trigger the changes in product's use, as well as discover co-causal relationships such that two or more different causes collaboratively "co-cause" a certain effect. We conduct extensive experiments to demonstrate the effectiveness of our approach on the Amazon Product Review Dataset that spans over 18 years. The proposed methods can be flexibly applied over different levels of product categories as well as they are generic enough to be utilized in other scenarios besides the studies of the technology evolution.

ACKNOWLEDGEMENT

Although only my name appears on the cover of this dissertation, lots of people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and thanks to whom my Ph.D. experience has been one that I will cherish forever.

My deepest gratitude is to my supervisor, Professor Katsumi Tanaka. I have been very fortunate to have an excellent advisor who gave me sufficient freedom and trust to explore research with my own unique ideas, while at the same time guiding me whenever my steps faltered. Thanks to his generosity, I had plenty of chances to attend many national and international conferences, where I learned a lot and met many researchers and professionals, which made me to think I still have a long way to go on the path of research. He is also a critical person and very strict to me on the research. I still remembered once in my lab presentation, he raised questions from the first slide, although sometimes I was quite "irritated" by his countless questions, yet I have to admit that his constant asking again and again made me become stronger in criticizing myself at any time and never give up of thinking. His patience and support helped me overcome many crisis situations and finish this dissertation. I hope that one day I would become as good an advisor to my students as he has been to me.

My special thanks goes to my "second boss" Associate Professor Adam Jatowt. He has been always there to listen to me and give me advice, as well as accompany me to complete every challenge. I feel gratitude to all the difficulties and failures in the research we were faced to, which undoubtedly paved the way to the success. I am deeply grateful to him for the long discussions and countless revisions of the papers and thesis. Without his deep thought and extreme seriousness, I cannot have completed today's achievement. I used to play a joke on his perfectionist attitude, but the facts proved that he is right and made me learned this is the indispensable characteristic for a good researcher.

Besides my advisor, I would like to thank my thesis committee: Professor Masatoshi Yoshikawa and Professor Sadao Kurohashi for their insightful comments and encouragement, but also for the tough questions which incensed me to widen my research from various perspec-

tives. They have also been my research advisers and gave me many precious comments and high level advice, which helped me to think outside the box and enrich my ideas.

I would like to show my great appreciation to my other research adviser, Professor Yasushi Sakurai at Kumamoto University. He taught me the importance of "vision" in opening a new research topic, rather than falling into some trivial issue. I also learned a lot from his optimistic attitude to everything, which allows to keep enthusiasm to the research.

My sincere thanks also goes to Assistant Professor Makoto P. Kato. I often consulted him with technical details in the research. He was always patient to help me organize my thoughts and gave me many valuable comments.

I would like to thank Professor Osami Kagawa at Osaka Gakuin University, Associate Professor Hiroaki Ohshima, Associate Professor Yoko Yamakata, Assistant Professor Takehiro Yamamoto and Lecturer Yusuke Yamamoto for good comments and suggestions during the laboratory meeting.

My sincere thanks are given to Professor Keishi Tajima at Kyoto University, Professor Akiyo Nadamoto at Konan University, Professor Hiroshi Ishikawa at Tokyo Metropolitan University and Professor Yoshifumi Masunaga at Ochanomizu University for their precious comments and discussions during conferences.

I would like to also thank the co-author of some of our papers, Associate Professor, Sourav S. Bhowmick at Nanyang Technological University. Thanks for his effort and comments in writing good papers.

I want to thank secretaries of Tanaka Lab: Ms. Mie Ashiwa, Ms. Rika Ikebe, Ms. Masumi Shirakawa and Ms. Kaori Sato. Thanks for their hard working and kind support, I could focus on my research and step forward smoothly. I really appreciated Ms. Ashiwa's help in daily life support as well as in translating the documents when I applied for research funding.

I thank my colleagues, especially Dr. Yoshiyuki Shoji, Dr. Kazutoshi Umemoto, Dr. Tomohiro Manabe, Ms. Meng Zhao, Mr. Rafael López García, Ms. Bei Liu and Mr. Zebang Chen for their kind help in research issues or in daily life. Thanks for all the lab mates for the stimulating discussions and for all the fun we have had in the last three years.

I appreciate the financial support from JSPS that funded parts of the research discussed in this dissertation.

Another special thanks must go to my adviser during my Master degree, Professor Daniel B. Neill at Carnegie Mellon University. Without his kind understanding and support, I would never have a chance to open up my story in Japan.

Finally, I would like to thank my parents Peng Zhang and Lu Ding for their deep love to me and their strong support to my every decision for the past 27 years. Lastly, thanks to my fiancé,

Lei Zhang, for his continuous and unfailing love, support, patience and understanding to make the completion of this thesis possible.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Text archives play a seminal role in preserving our cultural heritage and social opinions for future generations, and are prevalent in many areas including informatics, computational social science, finance, etc. Thanks to the improved digitization techniques, more and more documents including many past document are being archived and made publicly accessible. However, even though a lot of digitized documents are archived and made available, still there are many barriers when it comes to exploring and analyzing archival datasets.

Text Archive can be defined as a collection of electronic documents and other literary or language resources which have been created, collected and distributed for the purpose of various kind of researches or for preservation purposes. Text archive can have multiple "dimensions" according to which it can be analyzed, such as time, location, language, topical dimensions and so on. If we look at the archive through a particular dimension, then the archive can be regarded as arranged by the value on the dimension. For example, if we look into the temporal characteristics of an archive, all the historical records in the archive will be sequenced by the time points at which the documents were created. Similarly, if we regard the location as a key dimension, the documents in the collections can be segregated by the locations mentioned in each document or the locations where the documents have been created. In this thesis, we particularly focus on analyzing the text archive through the dimension of "time". In other words, each document in an archive is associated with the time point when this document was created and that time is a key aspect differentiating documents.

In the state-of-art studies on text archive, many researchers focused on character/named entity recognition in digitized archival data [37, 77, 69, 113], or on temporal analysis, such as how topics

evolve [3, 22, 118, 4], on the changes in named entities [70, 109] and the changes in terminology [7, 48, 49, 62, 47] in archival data. However, when it comes to the task of "search", there are few research works tackling with broader problems when users try to search for information in text archives.

One key problem lies in the difficulties of searching for object within document archives. Different from usual search conducted on the current web, the domain that an average, non-expert user wishes to access in archive data is often unfamiliar to her or him. For instance, let's assume a user wishes to search for documents describing the way people used to listen to music 200 years ago. Likely, he or she does not know the keyword "phonograph". In such a case, the current search engine based on a simple keyword marching may not output (at least, the sufficient amount of) relevant documents. This problem arises due to *terminology gap* which indicates the gap between the knowledge a user possesses and the text content he wishes to retrieve. This situation is usually the result of the limited knowledge of the unknown domain (e.g., the past) that average users possess, making them fail to select suitable key words for searching within the target content. It would be then beneficial if users were provided with some kind of assistance when interacting with archival document collections. Ideally, such assistance should empower them to search/explore within the unknown collections as efficiently as they would do in familiar collections such as the Web.

Besides the object search, the terminology gap also influences the tasks when users try to search for the relationships in another domain. For example, this happens when searching for a set of evidences to indicate or explain the similarity relationship among entities existing in different domains in the archive. The current similarity measures limit the comparison scope to either the entities which belong to the same domain by simply comparing their context terms, or to the entities in two different domains yet appearing in a structured data (e.g., a relational database) [52, 114]. The task of across-domain similarity relationship extraction from unstructured text archive (e.g., news) remains still open to be solved.

The cause-and-effect relationship is another major relationship which is likely to be searched by users who wish to discover and understand the changes that are represented in text archives. For example, a user may desire to search for the reason "why" the changes occur and the effects these changes may cause or trigger. *Causality detection* within text has been researched for many years in the area of Natural Language Processing (NLP), however, for longitudinal text archives, standard Natural Language Processing (NLP) methods [35, 34, 78, 88] designed for causality or entailment detection within text cannot be directly applied. These can extract only explicitly mentioned causal relations, while in our work, we aim to detect implicit causality that relates temporally distant events.

To sum up, we make the following contributions for providing better functionality for searching in and analyzing text archives:

1. We propose a general and unsupervised transformation technique to bridge the terminology gap across different domains, which provides fundamental support for the tasks of object search as well as relationship search in text archives.

2. We describe several similarity measures to enable estimating the similarity relationship between entities from different domains in document archives.

3. We introduce a novel approach for implicit causal relation extraction to explain the changes occurring in temporal document collections. The proposed methods can be flexibly applied over any time-stamped text document archives.

1.2 Approaches

1.2.1 Overview

In general, this thesis mainly tackles with two tasks for providing better user search experience in text document archives: (1) *object search* and (2) *relationship search*. As for the *relationship search* task, we specifically focus on detecting similarity relationship and causal relationship. In such sense, we name the two sub-tasks as *evidence search* and *causality search*. We propose several methodologies to solve technical barriers related to the above-mentioned tasks: (1) transformation technique to combat the terminology gap, (2) novel similarity measures to enable effective estimation of the similarity relationship between across-domain entities and (3) implicit causal relation extraction to explain changes recorded in archives. Fig. 1.1 overviews the dependencies among all these tasks as well as the relationships between the tasks and methodologies used for solving them.

First, the tasks of *object search* and *relationship search* are related to each other. The former helps to quickly detect query's counterparts (equivalent entities to the query) in another domain (indicated as (a) in Fig. 1.1); one branch of the latter can provide detailed evidence to explain the similarity between query and its counterpart by *evidence search* (indicated as (c→b) in Fig. 1.1). Another branch, *Causality search*, enables to explain the evolving sequence among entities in text archive (indicated as (d→b) in Fig. 1.1). It relies on discovering cause-and-effect relations between objects, which may enrich the entity inter-relationships apart from the similarity relationships within entities.

As mentioned before, the terminology gap concerns fundamental problem in text archive search and analysis assuming domain difference (e.g., time distance between the creation time of

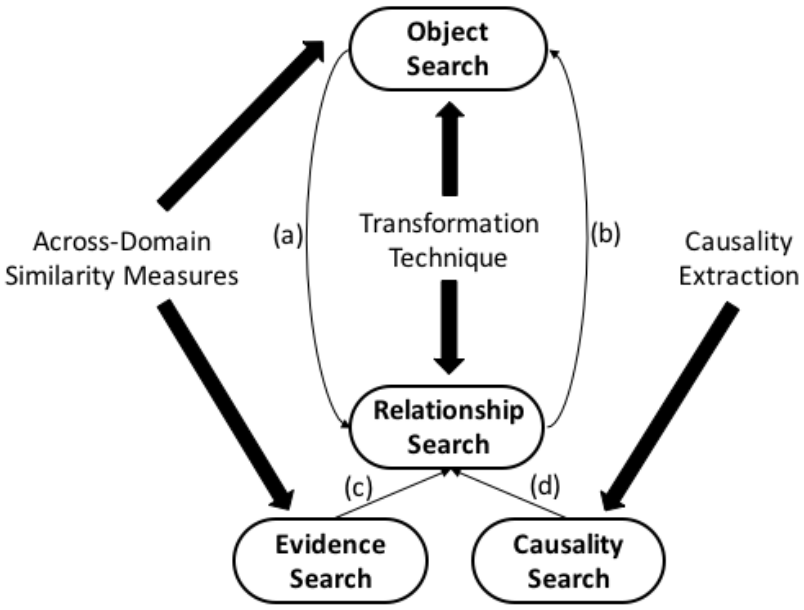


Figure 1.1: Overview of our approaches towards improving search and analysis of text archive.

documents) of different portions of the same archive or those of different archives. Hence, the transformation technique is of crucial importance and provides foundation for all the three tasks [1]. The across-domain similarity measure constructs the similarity between entities from different domains which assists searching counterparts and relation extraction across different domains. The technique of causality extraction serves for the task of *Causality Search*.

1.2.2 Object Search

To help users search in unknown domains, we allow them to input an entity (e.g., `iPod`) in their familiar domain (e.g., present times such as 2000s) and then we transform the queried entity into its counterpart term (e.g., `Walkman`) in the target unknown domain (e.g., some past time period such as 1980s). The key idea of transformation technique is to first bridge the two domains by its shared key structures as anchors. Then, once the cross-domain mapping has been found using the anchors, the other terms within the two spaces can be aligned by the similarity of their positions relative to the anchor terms in their own spaces. This transformation process is query independent and aims at building a global correspondence between the two domains. We call it *global transformation*. However, such global mapping can capture only common information or common facts neglecting ones particular to a given query, thus we also propose another method called *local transformation* to assure more precise term mapping by explicitly leveraging the relationships between the query and its informative context terms. As output

[1] Note however that in the current work we do not apply transformation technique for causality detection, it is regarded as key component to be included as future work

for both the methods, we provide users with a list of candidate counterparts in the target domain. These can be used for example as query suggestions shown to users to conduct further exploration in the text archive.

1.2.3 Evidence Search

This task aims at constructing similarity criteria and measures to enable effective entity comparison across different domains as well as at explicitly extracting evidence to explain and support similarity-focused reasoning of two arbitrary entities. Users are supposed to input two entities (e.g., *iPod* vs. *Walkman*) which belong to two different domains (e.g., 2000s and 1980s). The output is then the set of term pairs (e.g., {`music-music`, `portable-portable`, `Apple-Sony`, `MP3-tape`}) which denote commonalities and aligned differences between the queried entities. We propose several similarity measures, including semantic similarity, relational similarity as well as systematic-based similarity (i.e., one that considers structural similarity) between terms of an input entity pair. As mentioned above, these techniques can help to support search task by providing evidence to understand results of counterpart search. They can be also used to re-rank initial search results.

1.2.4 Causality Search

We approach the task of discovering cause-and-effect relationships by defining a causal relation within text archive as the relation between a cause term c and an effect term e, where the change in the cause leads to or triggers the change in the effect. Based on this definition, we first define the two types of changes: change in popularity described by term frequency (frequency-based) and change in semantic meaning (semantic-based). After detecting the changes, we propose several techniques to estimate the causal strength between the cause term and the effect term which are terms undergoing such changes. Besides the discovery of binary causal relations between two single terms, we further propose different aggregation processes for grouping binary causal relationships to provide more evidence for supporting causality selection as well as to improve precision of the returned results.

1.3 Thesis Structure

In Chapter 2, we survey prior work related to the research problems presented in this thesis. Every following chapter after Chapter 2 in this thesis corresponds to a particular research task among those shown in Fig. 1.1:

- Chapter 3
  We address the problem of terminology gap in object search by detecting semantically similar terms across knowledge domain. For example, users searching for documents about

music devices in 1980s which are similar to iPod, may fail to succeed since they do not know the keyword Walkman which played similar role as iPod does nowadays. So we solve this problem by finding the counterpart (e.g., `iPod`) of the query (e.g., `Walkman`) that existed in the target domain (e.g., 1980s). We propose an efficient method to find counterparts by transforming the representation of terms within different vector spaces. We then enhance the global correspondence method by considering also the local context of terms (local correspondence). Through the extensive experiments on the a 20-years long archive, the New York Times Annotated Corpus, as well as a 200-years long archive, the Times Archive, the proposed methods are proved to be effective in searching for equivalent entities of different types of input, queried entities (objects, persons and locations).

- Chapter 4

  We introduce in this thesis a novel problem of automatically explaining similarity of entities in archives. We propose an effective approach for this task, which, in a fully unsupervised fashion, returns a set of evidence indicating similar aspects of entities dispersed across different domains. We detect their commonalities as well as aligned differences by proposing two different approaches for selecting explanatory terms. We then evaluate the proposed methods on the New York Times Archive and demonstrate that they can successfully detect both commonalities and aligned differences of compared entities. To better present the results discovered in these two topics, we also introduce two views for result investigation: the first one, called the top counterpart view, visualizes the top temporal counterparts of a queried entity, while the second one, the similarity explanation view, displays the extracted evidences to support the understanding of similarity between the query and a selected counterpart.

- Chapter 5

  In this chapter we investigate how the objects change and what are the effects of these changes by detecting causal relations in a specific type of text archive, online product reviews. We are particularly interested in understanding social impact of technology and in discovering how changes of product features influence changes in our social lives. We address this problem by providing novel methodology to extract evidences of technology impact on human activities. In particular, we first distinguish two kinds of product-related terms: physical product features to represent the potential causes and terms describing situations when products are used as the potential effects. We then detect changes in both types of terms over time by tracking fluctuations in their popularity and usage. Finally, we discover cases when changes of product features trigger the changes in product's use, as

well as we find co-causal relationships such that two or more different causes collaboratively "co-cause" a certain effect. Extensive experiments are then conducted to demonstrate the effectiveness of our approach on the Amazon Product Review Dataset that spans over 18 years. The proposed methods can be flexibly applied over different levels of product categories as well as they are generic enough to be utilized in other scenarios besides the studies of the technology evolution.

- Chapter 6
  The last chapter summarizes the thesis and addresses several directions to be explored as future work.

CHAPTER 2

RELATED WORK

2.1 Bridging Terminology Gap across Domain

Temporal changes in word meaning have been an important topic of study within historical linguistics [2, 17, 63, 45]. Some researchers employed computational methods for analyzing changes in word senses over time [72, 57, 47, 62]. For example, Mihalcea and Nastase [72] classified words to one of three past epochs based on words contexts. Kim et al. [57] and Kulkarni et al. [62] computed the degree of meaning change by applying neural networks for word representation. Jatowt and Duh [47] used sentiment analysis and word pair comparison for meaning change estimation. Our objective is different from the one in those works as we search for corresponding terms across time, and, in our case, temporal counterparts can have different syntactic forms.

Some work have already approached the problem of computing term similarity across time [7, 48, 50, 109]. Berberich et al. [7] first proposed to solve this problem by using an HMM based model and measuring the across-time semantic similarity of two terms to compare their contexts as captured by co-occurrence measures. We compare their work with our approaches due to the same objective and and the same dataset (NYT) used, as well as due to its flexibility in querying counterparts over arbitrary two time periods. Kalurachchi et al. [48] approaches the across-time similarity finding task by checking if entities (nouns) are referred to using the same verbs in different time periods. In our approach we do not assume the existence of effective POS taggers for historical texts, which can be of various genres, and which generally tend to contain many errors due to OCR process, especially, for longer distances. The objective of the other related work is different from ours. For example, Tahmasebi et al. [109] attempts to detect the name changes of an input entity (e.g., *St. Petersbourg-Leningrad*) by analyzing the contexts during the

change periods to find the temporal co-references of different names. In this aspect, their method can be applied to the cases when temporal counterpart is the same entity, albeit called differently at different times. However, their method is not applicable for detecting two entities which are semantically similar, yet, which have different identities (e.g., *iPod* and *Walkman*). Finally, the approach of Kanhabua et al. [49] is corpus-specific (based on the Wikipedia snapshots) and thus cannot be directly applied for the case of unstructured temporal archives spanning long time periods.

The idea of distance-preserving projections across domain is also used in automatic translation [75]. Our research problem is however more difficult and is still unexplored. In the traditional language translation, languages usually share the same concepts, while in the across-time translation concepts evolve and thus may be similar but not always same. Furthermore, the lack of training data is another key problem.

Transfer Learning [83] is related to some extent to our work. It has been mainly used in tasks such as POS tagging [12], text classification [11, 67, 115, 117], learning to rank [16, 28] and content-based retrieval [52]. The across-domain correspondence problem can be also understood as a transfer learning as it is a search process that uses samples in the base domain for inferring correspondent instances in the target domain. However, the difference is that we do not only consider the structural correspondence but we also utilize the semantic similarity across time.

2.2 Entity Comparison across Domains

Analogical relation detection [24, 112, 111] is related to our work. Structure Mapping Engine (SME) [24] was the original implementation of Structure Mapping Theory (SMT) [29] that explains how humans reason with analogy. Later, Turney proposed the Latent Relational Mapping Engine (LRME) [112] that extracts lexical patterns in which words co-occur to measure relational similarity. The problem of explaining the analogy was however never explicitly approached. Note that in the case of the across-time comparison we cannot assume that users know commonalities and aligned differences of entities at different times, especially ones from distant past. Thus, the explanation task is clearly needed. Furthermore, our task is more difficult than the analogy detection due to the need of considering meaning changes of terms over time.

Some previous work dealt with the task of finding similar terms across time [7, 49, 48, 119]. Our objective is different as we attempt to detect explanatory terms to reveal similarities between two entities rather than to find temporal counterparts of a single entity.

Lastly, this work could be also considered as related to comparative summarization [114] which requires finding difference information within input documents. However, comparative summarization does not explain similarities of entities. Furthermore, our proposed methods work

on unstructured texts (e.g., news articles) which naturally do not exhibit the same characteristics as data in structural datasets such as in relational databases.

2.3 Causality Detection

Causal relations can be crudely classified into type-level causality and token-level causality. The type-level causality is usually based on periodical or recurring causal relations (e.g., low air pressure causing rain, or common cold causing runny nose). This kind of causality is detected by analyzing multiple instances of a given type (e.g., low air pressure in different locations followed by rain in these locations, or several patients diagnosed with common cold who later reported the running nose condition). On the other hand, in token-level causal relations the cause is specific and often occurs only once (e.g., the release of a new technology causing certain response of users). Due to the lack of directly applicable training data (i.e., multiple instances of the same type), in general, the token-level causality is more difficult to be detected. In this work, because of the character of the approached task, we detect the token-level causal relations.

A variety of approaches from computer science and statistics have been developed for detecting causal relations [59]. These can be categorized into NLP-based approaches [10, 19, 33, 35, 34, 42, 51, 53, 78, 88], Graphical models [6, 15, 21, 27, 46, 79, 103, 107], Granger causality [5, 31, 36] and temporal logic based ones [39, 54, 60].

Most of the causality works in the area of NLP are based on pattern extraction [10, 19, 35, 78, 88]. For example, Girju et al. [35] propose detecting causality by first discovering the lexico-syntactic patterns in text that refer to causation and then applying semantic constraints to validate and rank the candidate patterns using the confidence scores generated by these constraints. Other research works such as [78, 88] focus on the problem of predicting the causality between events by generating semantic rules to express the causality in text. However, the limitation of the semantic rule (or pattern) based approach is the difficulty of extracting all possible ways in which users could express causality. Another limitation of this approach is that it can extract only explicitly mentioned causal relations in text, while in our work we attempt to detect implicit causality that relates temporally distant events, and that may not be directly referred to in text.

Graphical models techniques (e.g., Bayesian Networks or Dynamic Bayesian Networks) infer probabilistic structures representing the set of causal relationships between given variables. Bayesian networks can be divided based on the way of constructing the best graph. One category of approaches relies on searching over the set of possible graphs to maximize a scoring function [21]. The other one starts with a fully connected graph and removes the edges by conducting conditional independence tests [107]. The limitations of graphical models are as follows. They require significant computation resources, which may make it impossible to exhaustively search

over all possible graphs. They also cannot test complex relationships. Regarding the latter issue, for example, within the graph we cannot determine whether two nodes A and B together will cause node C at a specific time T.

Granger causality [36] is a popular approach for causal relation detection that has roots in economics and econometrics. The key idea underlying Granger causality is to perform hypothesis tests for every pair of variables in order to test whether some time series are predicative about other time series with a given lag. To check if a variable A "Granger causes" B we need to examine if A provides extra explanatory power to predict B which is higher than the one obtained from the past values of B itself. However, this approach has several drawbacks. It involves applying regression on lags resulting in O(p2) time, where p is the number of features. Also, the statistical significance tests are conducted sequentially for all the pairs of features. Consequently, several extension works based on the concept of Granger causality have been proposed to tackle the complexity problem, such as Lasso Granger method [5], Vector Auto-Regressive (VAR) [31], etc. Another drawback is that Granger causality based methods as well as Bayesian Nets typically solve the problem of type-level causality, which targets periodical or general causal relations (e.g., low air pressure causing rain, or common cold causing runny nose). However, our goal is to detect token-level causal relations where the cause is specific and often occurs only once (e.g., the release of a particular new technology causing certain user response). In general, token-level causality is more difficult to be detected than type-level causality due to the lack of direct prior evidences.

Finally, solutions using temporal logic are based on the assumption that a cause is earlier than its effect and the cause raises the probability of the effect [39]. Unlike the above-mentioned approaches, the temporal logic can be employed for analyzing the token-level causality. We thus adopt the temporal logic approach in our work thanks to its capability of handling the token-level causality. In our scenario we need to detect token-level causal relations occurring at a specific time period (in the form of innovation-related words) without prior evidence that is usually available for type-level causal relations. However, unlike the previous solutions based on temporal logic approach [39, 54, 60], our method is novel as (1) it detects implicit causal relations between words within temporal document collections. (2) Despite detecting the causal relations between two words, we also provide aggregation ways to better estimate the causal strength by grouping similar concepts or similar causal patterns.

CHAPTER 3

DETECTING SEMANTICALLY SIMILAR TERMS ACROSS DIFFERENT DOMAINS

3.1 Introduction

Our knowledge of the past tends to be limited. An average person typically knows only about events and entities which were taught at school or ones curated as collective memory - the highly selective representation of the past maintained by mass media. Searching within past collections as well as understanding the retrieved content can be then hampered due to our limited knowledge of vocabulary used in the past and its meaning. "The past is a foreign country: they do things differently there", one could cite here the opening sentence of a famous novel "The Go-Between" by L. Hartley. Considering the limited knowledge of the past the average users possess, it would be then beneficial if the users were provided with some kind of assistance when interacting with archival document collections. Ideally, such assistance should empower them to search within the past collections as efficiently as they would do in "present" collections such as the Web.

We focus in this work on solving the temporal counterpart search which, based on a longitudinal document collection, requires returning semantically similar terms from the past to an input query from the present time. For example, for a query *iPod* the system should return the ranked list of words used in the past that refer to objects (e.g., *Walkman*) having similar functionality and similar role as *iPod* has nowadays. Users present-ed with the ranked lists of temporal counterparts could then update and improve their queries for more effective search experience. In another scenario, users could receive semantically corresponding words from the present when encountering unknown words in browsed past documents (problem similar to text translation). The latter scenario means that effective approach should be independent of time orientation. In other words, our method should be also able to output present words for past words used as

queries.

Temporal counterpart detection is not trivial. Typically, there is low overlap between contexts of temporal counterparts [119] suggesting that direct context comparison will not work. We then make use of distributed word representation [74, 76] that allows decreasing the dimensionality of word meaning representation and improving the sparseness problem typical for vector space based word representations. Given the representations trained on the temporally distant time scopes (typically, the representation derived from the present documents and the one derived from the documents published at some period in the past), matching between words across time becomes possible through some kind of transformation. This essentially means aligning relative positions of terms in the corresponding vector spaces of different time periods using a transformation matrix. However, the inherent problem behind such an approach is difficulty of finding large training sets given the variety of domains, document genres and arbitrary time periods in realistic scenarios of temporal counter-part detection. Thus to design a robust method we pro-pose using automatically derived training sets to construct the transformation matrix for a given pair of time periods (called the base and target time periods). In particular, we introduce two approaches for selecting seed pairs of temporal counterparts to be used for training the transformation matrix. The first one assumes that frequent and common terms in both the time periods (i.e., the base and target time periods) can be used for optimizing the transformation matrix. This reasoning is based on the observation that frequent words are known to change their semantics across time only to a small degree [65, 84]. The second method extends the first one by using terms that are computationally verified to undergo little semantic variation across time.

The above approach allows creating mapping between terms across time. However, the transformation that it is based on, is the same for every term (hence, it is called a global correspondence) and may not perform optimally for any arbitrary query. To improve the temporal counterpart search we then also introduce an extended method called local correspondence that locally constrains a query by transforming its core context terms, which are automatically detected and treated as reference points. Both the approaches generate ranked lists of temporal counterparts as results.

Finally, we propose to enhance results of the above methods by outputting evidences which explain why particular term should be considered as a temporal counterpart of a given query. This is done by selecting pairs of terms that portray different similarity aspects between the query and the selected counterpart term. We next visualize both the top candidates of temporal counterparts and their detected similarity evidences on two dimensional pane using principal component analysis.

We perform experiments on the New York Times (NYT) Annotated Corpus [104], as well as

on the Times Archive . The latter collection is much larger than the former and, what is especially important, it has significantly longer time span stretching over to the 19th century. This allows us to test our methods on both short and long time frames. The experiments demonstrate that the proposed methods outperform the best performing baseline by 113% for NYT corpus, and by 28% for the Times Archive for queries that have different literal forms from their temporal counterparts and when OCR driven errors are corrected.

3.2 Problem Definition

In this section, we formally define the problem of temporal counterpart detection.

DEFINITION 1 TEMPORAL COUNTERPART: *If e is semantically similar to q in t_t, then e is temporal counterpart of q in t_t.* For finding temporal counterparts we base our approach on distributional hypothesis [40] which states that similar terms appear in similar contexts. Below we introduce several terminologies necessary for describing our approach.

DEFINITION 2 CONTEXT COMPARISON: *Let $C_q = \{x_1, ..., x_u\}$ be the set of context terms of q in t_b, and $C_w = \{x_1, ..., x_v\}$ be a set of context terms of e in t_t.* If C_e is semantically similar to C_q, then e is determined to be the temporal counterpart of q.

DEFINITION 3 SEMANTICALLY STABLE TERMS: *Semantically stable terms are terms that do not change their meaning over time. A semantically stable term x satisfies $x(t_b) \approx x(t_t)$.* Music is an example of a semantically stable term since *music*(2000s) $\approx$ *music*(1980s).

DEFINITION 4 SEMANTICALLY UNSTABLE TERMS: *Semantically unstable terms are terms that change their meaning over time. A semantically unstable term x satisfies $x(t_b) \neq x(t_t)$.* For example, *gay* and *mp3* are semantically unstable terms since *gay*(2000s) $\neq$ *gay*(1950s) and *mp3*(2000s) $\neq$ *mp3*(1980s).

PROBLEM STATEMENT Given a query term q associated with the base time t_b, and the target time t_t, the task is to find the temporal counterpart e of q that existed in t_t. The relation between q and e is:

$$q(t_b) \approx e(t_t)$$

CONCEPTUAL DESCRIPTION OF THE PROPOSED APPROACH . Semantically unstable terms can be represented by semantically stable terms such as ones selected from their surrounding terms. The meaning of a semantically unstable term can be then represented by a group of semantically stable terms that occur in its context, such that $x^{un}(t_b) = F(x^{s_1}(t_b), ..., x^{s_k}(t_b))$ where x^{un} is a semantically unstable term, x^s denotes a semantically stable term, and F is a nested representing function in time t_b (see Fig. 3.1 for visualization).

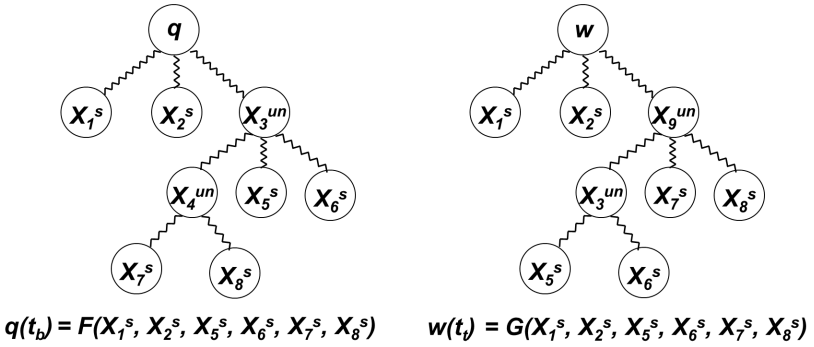


Figure 3.1: Query term and its temporal counterpart represented by semantically stable terms.

Based on this idea, we can rephrase the query context such that $C_q = \{X^s(t_b), X^{un}(t_b)\} = \{x^{s_i}(t_b), ..., F(x^{s_j}(t_b), ..., x^{s_k}(t_b))\}$ where X^s denotes semantically stable terms and X^{un} is the set of semantically unstable terms. X^{un} can be actually represented by a function of stable terms. Similarly, in the target space $C_w = \{X^s(t_t), X^{un}(t_t)\} = \{x^{s_i}(t_t), ..., G(x^{s_j}(t_t), ..., x^{s_k}(t_t))\}$ where G is a nested representing function in t_t. The measurement of the across-time similarity resolves to finding a mapping function between function F and function G, denoted as $\boldsymbol{M}(F, G)$.

3.3 Global Transformation

We focus in this section on constructing the mapping function between the base time and the target time. This process is query independent and can be done offline before a user issues a query. We first introduce the way to represent terms in the base time and in the target time within their respective semantic vector spaces, χ^B and χ^T. Then, we construct a transformation matrix as a general mapping function to bridge the two vector spaces.

3.3.1 Vector Space Word Representation

Distributed representation of words by using neural net-works was originally proposed by [102]. Mikolov et al. [74, 76] improved such representation by introducing Skip-gram model based on a simplified neural network architecture for constructing vector representations of words from unstructured text. Skip-gram model has several advantages: (1) it captures precise semantic word relationships; (2) it can easily scale to millions of words. After applying the Skip-gram model, $m \times p$ matrix is created from the documents in the base time, $D(T^B)$, where m is the vocabulary size and p are the dimensions of feature vectors. Similarly, $n \times q$ matrix is constructed from the documents in the target time, $D(T^T)$.

3.3.2 Transformation across Vector Spaces

Our goal is to compare words in the base time and words in the target time in order to find temporal counterparts. However, it is impossible to directly compare words in two different semantic vector spaces as the features (dimensions) in both spaces have no direct correspondence between each other. We then propose to train a transformation matrix in order to build the connection between different vector spaces. To better imagine the transformation idea, the semantic spaces could be compared to buildings. If we regard two semantic spaces as two buildings, then, in order to map the components from one building to ones in the other one, we need first to know how the main frames of the two buildings correspond to each other. Afterwards, the rest of the components can be mapped automatically by considering their relative positions to the main frames of their building. So, in our case, having found the correspondence between the anchor terms in the two semantic spaces, we can automatically map all other remaining terms relative to these anchors. Here, the anchors can be understood as semantically stable terms as defined in Sec. 3.2, while the other parts of the building can be regarded as semantically unstable terms. Fig. 3.2 conceptually portrays this idea by showing that the correspondence of anchor terms enables mapping other terms, such as *iPod* to *Walkman* so that the relative position between *iPod* and anchors in the base space is similar to the relative position between *Walkman* and the corresponding anchors in target space (only two dimensions are shown for simplicity).

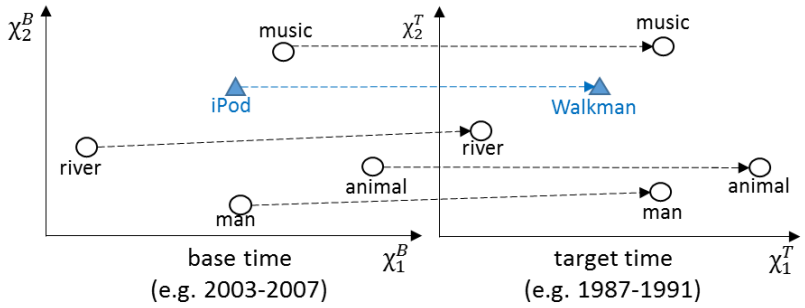


Figure 3.2: Conceptual view of the across-time transformation by matching similar relative geometric positions in each space.

For realizing the above described transformation, it is essential to first find good anchor terms ("main frames") which can help to build the correspondence between any two semantic spaces. However, manually preparing large enough sets of anchor terms that would cover various domains as well as exist in any possible combinations of the base and target time periods requires much effort and resources. We then rely here on an approximation procedure for automatically providing anchor pairs. Specifically, we select terms that (a) have the same syntactic forms in the base and the target time periods, and (b) are frequent in both the time periods. Such Shared

Frequent Terms (*SFTs*) are then used as anchor terms. One reason to use *SFTs* as anchors is that the frequent terms tend to be "connected" with many other terms. Another is that frequent terms (e.g., sky, river, music, cat) change their meanings over time only to small extent. The more frequently a word is used, the harder is to change its dominant meaning (or the longer time it takes to make the meaning shift) as the word is commonly used by many people. The phenomenon that words used often in everyday language had evolved more slowly than words used less frequently has been observed on several languages including English [65, 84]. This assumption should guarantee relatively good correspondence between the two frames that is "constructed" by *SFTs*. After determining the anchor terms, our task is to build the correspondence between two semantic spaces by utilizing the set of prepared anchor terms. In particular, we will train the transformation matrix to automatically map dimensions of the base vector space to the ones in the target vector space. Let us suppose there are K pairs of anchor terms $\{(\omega_1, w_1), ..., (\omega_k, w_k)\}$ where ω_i is a base time anchor and wi is its counterpart anchor in the target time. The transformation matrix $\boldsymbol{M}$ is then found by minimizing the differences between $\boldsymbol{M}\cdot\omega_i$ and w_i (see Eq. 3.1). This is done by ensuring that the sum of Euclidean 2-norms between the transformed query vectors and their counterparts is as small as possible when using K anchor pairs. Eq. 3.1 is used for solving the regularized least squares problem ($\gamma = 0.02$) with the regularization component added to prevent overfitting.

$$M = \underset{M}{\operatorname{argmin}} \sum_{i=1}^{k} \left\| M \cdot x_i^A - x_i^B \right\|_2^2 + \gamma \left\| M \right\|_2^2 \tag{3.1}$$

Note that the number K of *SFTs* used as anchor pairs is decided experimentally. In Sec. 3.7.4 we show the results achieved by using different numbers of *SFTs*. Having obtained matrix $\boldsymbol{M}$, any query q can then be associated with its corresponding temporal counterpart by first multiplying its vector representation with the transformation matrix $\boldsymbol{M}$. Next, the transformed query vector is compared using the cosine similarity measure with the vectors of terms in the target time. The most similar terms can be then returned as results. The across-time semantic similarity between the input term q in the base time and a given term e in the target time is defined as follows.

$$S_{sim}(q, e) = cos(\boldsymbol{M} \cdot q, e) \tag{3.2}$$

The idea of distance-preserving projections is also used in automatic language translation [75]. Note however that our research problem is more difficult and unexplored. In the traditional language translation, languages usually share same concepts, while in the across-time translation concepts evolve and thus may be similar but not always same. Furthermore, the lack of training

data is another key problem.

3.4 Global Transformation with Semantic Stability

In Sec. 3.3.2 we assumed that the frequent terms can be regarded as anchors because their meanings tend to remain relatively stable over time. To further ensure that the selected terms did not change their semantics across time (or changed to small extent), we introduce here additional method to track changes in word meaning. The purpose is to select only those frequent terms whose semantics are the least changing.

Our goal is to quantify how much the context of a given term evolves over time. We first collect all the words that occur in the corpus and treat them as combined vocabulary (regardless of time when these terms appeared). Based on such vocabulary we then train the Skip-gram model for each consecutive unit time by utilizing the portion of the corpus that consists of documents published in this time unit [1]. The settings here are such that from the beginning (i.e., the first time unit) every word is going to have a certain position in the vector space. Note that, for the terms which have not appeared in the first time period, the model still assigns some initial vectors which will be subsequently updated once the term starts to occur in the later time units. For each consecutive time unit [2], we iterate over epochs and train word vectors until convergence. The convergence is determined by comparing the average angular change in word vectors between epochs same as in [62]:

$$\rho = \frac{1}{|V_t|} \sum_{w \in V_t} arccos \frac{\chi_w(t, \varepsilon) \cdot \chi_w(t, \varepsilon - 1)}{\|\chi_w(t, \varepsilon)\|_2 \, \|\chi_w(t, \varepsilon - 1)\|_2} \tag{3.3}$$

where the $\chi^w(t, \varepsilon)$ is the vector of word w at time unit t and epoch ε. For each time unit t, after each epoch, the model stops updating the word vectors if ρ is lower than 0.0001.

We finally construct the time series of a word w by computing the semantic distance between its distributed representations at time t and at time $t - 1$. The semantic distance is measured as follows [3]:

$$Dist_t(w) = 1 - \frac{\chi_w(t) \cdot \chi_w(t - 1)}{\|\chi_w(t)\|_2 \, \|\chi_w(t - 1)\|_2} \tag{3.4}$$

This technique is then applied to the framework of the global transformation described in Sec. 3.3.2 by using anchors which are both frequent in the target and base time periods and which are

[1] Note that this training is independent from the one described in Sec. 3.3.1

[2] In the experiments we use one year as a time unit for the dataset New York Times.

[3] Note that there is no need for applying the space transformation in this case as term vectors at each time unit are computed by retraining data from the previous time unit.

also characterized by high semantic stability between these time periods. The semantic stability is computed as below. The parameter T denotes the time span during which the stability of w is tested:

$$Stability_T(w) = \sum_{t \in T} Dist_t(w)^{-1} \tag{3.5}$$

The following are examples of semantically stable terms: hotel, stores, opera, patients, disease, blood, songs and catholic.

3.5 Local Transformation

The methods described in Sec. 3.3 and 3.4 compute "global similarity" between terms across time. Since these are query independent measures, the transformation matrix (Eq. 3.1) is optimized over all the anchor points. Such global mapping can capture only common information or facts neglecting ones particular to a given query. Thus, it may underperform for some queries. For in-stance, when the query is `iPod`, the global transformation will output many electronic devices, such as `VCR`, `Macintosh`, `powerbook` as its top counterparts in 1980s. `VCR` is found to be the temporal counterpart of `iPod` merely because both of them possess recording as well as playback functions. On the other hand, `Macintosh` is determined to be corresponding to `iPod` since both seem to be manufactured by `Apple` company. Although, `VCR` and `Macintosh` are related to `iPod`, one cannot say they are its correct temporal counterparts. As we can see the global transformation may fail to output correct counterparts since it neglects key relations between a query term and its context.

Inspired by these observations, we propose another method to assure more precise term mapping by explicitly leveraging the relationship between the query and its informative context terms called *reference points*. The reference points serve as local anchors for mapping the query with its correct temporal counterpart by utilizing their relations with both the query and temporal counterpart candidates. We call such similarity matching *local transformation*. In the following subsections, we will describe the way to detect *reference points* and the way to find temporal counterparts using the detected reference points.

3.5.1 Reference Points Detection

Reference points are terms in the query's context, which help to build connection between the query and its temporal counterparts. Reference points should have high relation with the query and be sufficiently general. We propose to use hypernyms of terms to satisfy the above requirements. General terms are preferred rather than specific or detailed ones since the concept behind a general term is more likely to persist across time, while detailed or specific terms are less likely

to have corresponding terms in the target time. To detect hypernyms in an efficient way, we adopt the method proposed by Ohshima et al. [82] that uses bi-directional lexico-syntactic patterns. This approach is characterized by high speed and it does not require any external ontologies. The latter advantage is important since there are no ready resources such as WordNet for arbitrary periods in the past.

Note that in our previous work [119], we have tested three different types of reference points: (1) frequent co-occurring context terms, (2) context terms which are semantically diverse from each other and (3) hypernyms of terms. We have found that hypernyms perform best. This is because other approaches ((1) or (2)) tend to select very specific terms from among the frequent co-occurring terms (such as {*iTunes*, *digital*} for *iPod*), and the concepts of these words may not have existed in the past.

3.5.2 Local Graph Matching

We represent the task of the counterpart selection as a graph matching problem in which the query-related graph is compared to corresponding graphs built based on the selected context terms of candidate counterparts.

Formulation. The local graph S_q of a query q is composed of a set V of vertices, such that $V = q \cup F$, where $F = \{f_1, f_2, ..., f_u\}$ is the set of reference points, and of the set of edges $\Pi = \{\pi |$ edges between q and each node in $F\}$. u denotes the number of reference points which is set up to 5 by default Our objective is to find a graph $S_e = (V', \Pi')$ in the target vector space that is most similar to S_q. $V' = e \cup F'$ is the set of vertices in this graph consisting of the temporal counterpart candidate e and its corresponding reference points $F' = \{f_1{}', f_2{}', ..., f_u{}'\}$. Finally, Π' is the set of edges between e and each node in F'.

Graph similarity computation. To compute similarity between graphs generated from different vector spaces (i.e., the query graph and the candidate counterpart's graph), we measure both their across-time *semantic* and *relational similarities*. Fig. 3.3 conceptually portrays this idea.

Semantic similarity is defined as the similarity between the nodes in the graphs (similarity of a node in query graph to the node in the candidate counterpart's graph) and it ensures that the compared nodes in the two graphs are semantically similar. We use Eq. 3.2 to compute across-time semantic similarity between the nodes.

Relational similarity quantifies the similarity between two relations (i.e., edges) across-time. Its objective is to measure the similarity degree of the relative positions of terms (nodes) with respect to the query and the candidate counterpart. *Relational similarity* is computed as follows.

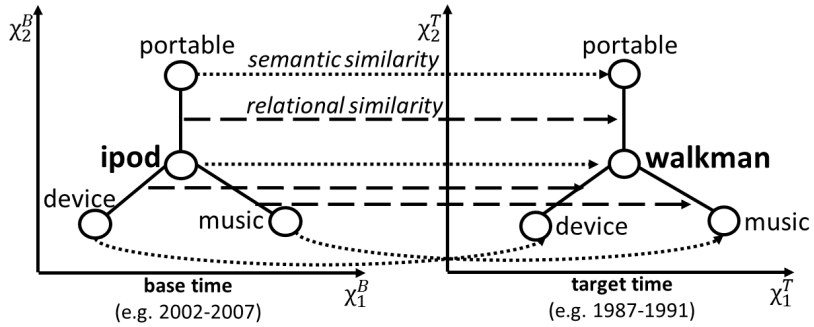


Figure 3.3: The concept of computing semantic and relational similarity in matching local graphs.

$$R_{sim}\langle(q, f_i), (e, f_i{}')\rangle = cos(\boldsymbol{M} \cdot (q - f_i), (e - f_i{}')) \tag{3.6}$$

The way to compute the similarity between the query graphs S_q and the candidate?fs graph S_e is then:

$$\begin{aligned} g(S_q, S_e) &= \lambda \cdot \sum_{i=0}^{u} S_{sim}(v_i, v_i{}') + (1-\lambda) \cdot \sum_{i=1}^{u} R_{sim}\left\langle(v_0, v_i), (v_0{}', v_i{}')\right\rangle \\ &= \lambda \cdot S_{sim}(v_0, v_0{}') + \sum_{i=1}^{u} h(v_i{}') \\ where, h(v_i{}') &= \lambda \cdot S_{sim}(v_i, v_i{}') + (1-\lambda) \cdot R_{sim}\left\langle(v_0, v_i), (v_0{}', v_i{}')\right\rangle \end{aligned} \tag{3.7}$$

Eq. 3.8 is the initial optimization function while Eq. 3.9 shows the derived function for more efficiently finding the best corresponding graph (to be explained later).

$$S_e^* = \underset{S_e \subset Vocab, |S_e|=u+1}{\operatorname{argmax}} g(S_q, S_e) \tag{3.8}$$

$$\begin{aligned} S_e^* &= \underset{S_e \in X}{\operatorname{argmax}} (\lambda \cdot S_{sim}(v_0, v_0{}') + \sum_{i=1}^{u} h(v_i{}')) \\ where, X &= \{\{v_0{}', v_1{}', ..., v_u{}'\} | v_0{}' \in Vocab, v_i{}' = \underset{v_i{}' \in Vocab}{\operatorname{argmax}} h(v_i{}')\} \end{aligned} \tag{3.9}$$

v_0 and $v_0{}'$ represent here the query q and its counterpart e. v_i and $v_i{}'$ denote the reference points and their correspondence in another space. Parameter λ is used for determining the degree to which the semantic or relational similarities are used (it is set to 0.5 by default). In Eq. 3.8 and Eq. 3.9, Vocab represents the entire vocabularies in the target space.

Time Complexity. Let N be the vocabulary size in the target time and let the base graph S_q contain u reference points as well as a query q. Then, naturally, in the target time, there are in total O($N^(u + 1)$) combinations of graphs that need to be compared in order to find the graph

Algorithm 1 Local Graph Matching

Input: Local graph of q, $S_q^{F_B}$.
Output: ranked list of candidate temporal counterparts E.

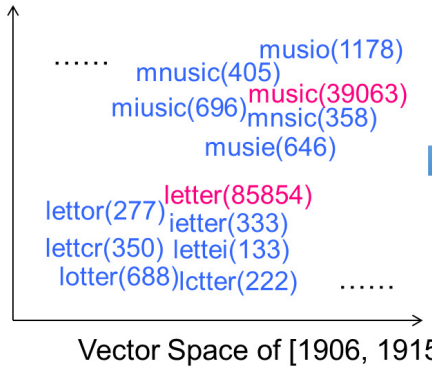
$E \leftarrow$ top k corresponding terms of q (by Eq. 3.2)
$FF \leftarrow \{$top k corresponding terms of each f in reference points $F_B = \{f_0, f_1, ..., f_u\}\}$ (by Eq. 3.2)
for each $e \in E[1:k]$ **do**
  $sum\_cos = 0$ # total graph similarity score
  for each $F \in FF[1:u]$ **do**
    $max\_cos = 0$ # current maximum graph similarity
    for each $f \in \mathcal{F}[1:k]$ **do**
      find f which maximizes current graph similarity (by Eq. 3.7)
    end for
    $sum\_cos+ = max\_cos$
  end for
end for
Sort E by sum_cos

with the highest similarity (see Eq. 3.8). This is prohibitively expensive to compute for the large size vocabulary (e.g., 300K-500K in a typical dataset). To solve this problem, we reduce the total number of combinations to $N \cdot N \cdot u$ by assuming the selection of each $v_i{}'$ be only dependent on the selection of $v_0{}'$ (e), yet, be independent on other $v_i{}'$ (see Eq. 3.9). Since N is still a large number, we further decrease N to k the most similar terms by applying Eq. 3.2 (by default, k is set to 1,000). Then the time complexity is changed to O$(k^2 \cdot u)$. Algorithm 1 summarizes this process.

3.6 OCR Error Correction

Document collections spanning long time periods such as Times Archive tend to suffer from OCR impreciseness. In the absence of manual verification of OCR results, the overall quality of documents is greatly impacted by numerous OCR errors, especially, in the case of older newspapers. For example, in Times Archive the words like *lettor*, *musio* or *motcr* commonly appear instead of their correct versions: *letter*, *music* and *motor*, respectively. To alleviate the impact of the OCR impreciseness on the performance of the proposed methods, we automatically construct a dictionary to map wrong word spellings to their correct forms, such as the case shown in Fig. 3.4. We note here that the previous research performed spelling checking mainly by using dictionary-based matching [13, 81, 101]. However, within longitudinal archives, vocabularies tend to change a lot. This situation makes it is difficult to prepare ready dictionaries for any required time period and particular document collection that would contain all the wrongly spelled words (especially, named entities) being mapped to their correct forms. Hence, instead, we em-

ploy here an approach that does not require any hand-crafted dictionary, while still performing reasonably well.



Original Spelling	Correct Form
mnusic	music
miusic	music
musie	music
.....	.....
lettcr	letter
lettor	letter
lotter	letter
.....	.....

Figure 3.4: Input and output of solving OCR problem (the numbers in the parentheses indicate the frequencies of terms in the Times Archive).

We first list assumptions underlying our approach:

1. *The misspelled term shares similar context with its correctly spelled form. Hence, if a word is misspelled then its correct form should appear within the set of its semantic neighbors. In other words, the wrongly spelled term will be positioned close in the vector space to its correct form.*

2. *The wrongly spelled term has similar literal form to its correctly spelled term.*

3. *The misspelled term is more dominant (or frequent) compared to its wrongly spelled variants.*

In general, we are checking the semantic neighbor set C of any word w (C contains the k-Nearest Neighbors of w where k is set to 5). If we find a semantically similar term c ($c \in C$) which meets the above conditions (it has one edit distance from w and it has higher frequency than w in the corpus), then we regard c as the correct form of w. Note that it is unlikely to have a wrongly-spelled term that is more frequent than its correct form (i.e., opposite situation to requirement 3 above). Most of OCR algorithms have over 70-80% accuracy rate or more. Even if there would be some occasional errors caused by our technique they are likely to be rare in the entire corpus and less than the number of correct changes. We then directly apply the constructed mapping dictionary to the obtained results (i.e., the ranked list of temporal counterparts) as post-processing to replace any incorrect spelling result with the correct one and, by this, also, to remove duplicate items.

3.7 Experiments Over Short Time Periods

In this section we describe the first part of our experiments conducted on relatively short time periods.

3.7.1 Datasets

We use here the New York Times Annotated Corpus [104]. This dataset contains over 1.8 million newspaper articles published between 1987 and 2007. We first divide it into four parts according to the articles?f publication time: [1987, 1991], [1992, 1996], [1997, 2001] and [2002, 2007]. Each time period contains around half a million news articles. We next train the model of distributed vector representation separately for each time period. The vocabulary size of the entire corpus (i.e., from 1987 to 2007) that we use is 476k after removing terms with frequency less than 5. Each time period has on average 337k (357k, 285k, 345k and 364k, respectively). We will first focus on the pair of time periods separated by the longest time gap, that is, [2002, 2007] treat-ed as the base time and [1987, 1991] used as the target time. We will then analyze the results when using more recent target time: [1992, 1996].

3.7.2 Test Sets

To the best of our knowledge, there are no benchmark datasets for our research task. We then resorted to manual construction of test sets. The test sets contain queries in the base time and their temporal counterparts in the target time. To create the test sets we consulted Wikipedia as the prime resource since it provided clear and unambiguous information on many entities including data on their lifetimes (e.g., time period in service of a prime minister or the time when the name of a country or of company changed). We have also used Web search engines or historical textbooks for confirming some of the answers. The test terms contain three types of entities: persons, locations and objects. Persons mainly include presidents, prime ministers or chancellors of the most developed and populous countries (e.g., `USA`, `UK`, `France`, etc.). We focused on such countries since they tend to be sufficiently frequently described, referred to or mentioned in our dataset. The person's list also included the names of popes and FIFA presidents. Locations included countries or cities (e.g., `Czechoslovakia`, `Berlin`) that changed their names over time, split into several countries, merged into another political system, or capitals that moved from one city to another. Finally, objects covered instances of devices (e.g., `iPod`, `mobile phone`, `dvd`), concepts (e.g., `email`), companies/institutions (e.g., `NATO`, `Boeing`) or other objects (e.g., `letter`, `euro`). The ratio of entities to non-entities is 7:3.

Note that a temporal counterpart does not need to be unique. For example, several terms such as `letter`, `mail`, `fax` are answers to the same query term, email, in 1980s. In total, 95 pairs

of terms (query and its counterpart) resulting from 54 input query terms are used for the task of mapping [2002, 2007] with [1987, 1991]. For testing the matching between [2002, 2007] and [1992, 1996] we use 50 term pairs created from 25 input query terms. The test pairs are publicly available at the specified URL [4].

3.7.3 Evaluation Measures and Tested Methods

Evaluation Measures. We use the Mean Reciprocal Rank (MRR) as the main measure for evaluating the ranked search results. MRR's values range between [0,1]. The higher the obtained value, the more correct the tested method is. Besides MRR, we also report precision @1, @5, @10 and @20. The precisions are equal to the rates of queries for which the correct counterpart term is found in the top 1, 5, 10 and 20 results, respectively.

Baselines. We prepare five baselines as follows:

(1) **Bag of Words approach (BOW)**: we test BOW approach to examine whether the distributed vector representation and transformation are really necessary. This method directly compares the context of a query in the base time with the context of a candidate term in the target time. To build the vector space under BOW approach, we utilize the top 5,000 frequent terms, called feature terms (excluding stopwords), as the dimensions of the vector space. This choice is motivated by efficiency reasons and by the observation that 5,000 most frequent terms can cover around 88.7% of text in a very diverse corpus of over 1,000k running words [80]. We represent each term v in the corpus by a feature vector. The values in this vector are obtained from the sentence-level co-occurrence counts of v with each of the feature term. We then scale these counts by the number of sentences containing a given feature term. The similarity between any two terms is then computed as cosine similarity between their feature vectors.

(2) **Latent Semantic Indexing (LSI) without transformation (LSI-Com)**: the purpose of using **LSI-Com** is to investigate the need for the transformation over time and the usefulness of neural network representation. To set this baseline we first merge the documents in the base time and the ones in the target time. Then, we train LSI [23] on the combined collection to represent each term by the same distribution of detected topics. Notice that the trained vector space contains both the vocabularies of the base time and the target time. We next search for terms in the combined vector space that are semantically similar to query by directly comparing their vector representations, and we select only the ones that existed in the target period.

[4] http://www.dl.kuis.kyoto-u.ac.jp/~adam/temporalsearch_short.txt

(3) **Latent Semantic Indexing (LSI) with Transformation (LSI-Tran)**: **LSI-Tran** is used to investigate if LSI can be an alternative for the neural network based vector representation based on the transformation scenario. We train two LSI models separately on the documents in the base time and the documents in the target time. Then we train the transformation matrix in the same way as we did for the proposed methods. We next compare the transformed vector representation of a given input query term with terms in the target time.

(4) **Skip-gram Model without transformation (W2V-Com)**: the purpose of including this baseline is similar to that of **LSI-Com**. Since **W2V-Com** also uses distributional representation for capturing word semantics as the proposed methods do, we can then evaluate the necessity of the transformation by testing this method in comparison to the proposed methods. The process is similar to the one for **LSI-Com** with the only difference that the vector representation is based now on the Skip-gram model.

(5) **Hidden Markov Model (HMM) proposed by Berberich et al. [7]**: the key idea behind this method is to measure the degree of across-time semantic similarity between any two terms by comparing their context words based on co-occurrence statistics. The across-time query reformulation technique utilizes a Hidden Markov Model. We select this approach as a baseline as (a) its objective is identical to ours (searching for temporal counterparts); (b) the input is unstructured data same as one used in our approach (i.e., unprocessed temporal news article archive such as New York Times dataset) unlike [49] which utilizes Wikipedia snapshots; (c) this method solves the general problem of temporal counterpart finding, hence, not only finding the name changes of the same entity.

Note that there is difference in the inputs. The available code that we could use has been designed to operate on year-long time periods. This means that both the tar-get and the based time periods should be single years. We then needed to adjust the results of year-to-year search (e.g., search from 2005 to 1990) of **HMM** [7] to handle period-to-period search type (e.g., search from [2002, 2007] to [1987, 1991]) used in evaluation [5]. For this we first ran **HMM** separately for every combination of a single year from the target time period (present time period) and single year from the base time period (past time period). We then aggregated the results for all the year combinations by either taking the minimum rank (i.e., the best position across all the returned rankings for all the year-to-year comparisons) of a given candidate counterpart or its average rank (i.e., the average position in all the rankings). The former approach is denoted **HMM_min**. It favors terms which achieved a good rank at least once across all the combinations of years from the target and the base time periods. The

[5] Document set from a single year is often not enough for neural network to sufficiently represent term meanings.

latter is called **HMM_avg** and it prefers terms consistently ranked at the top positions for different year combinations.

Proposed Methods. We test all the proposed methods mentioned in the previous sections. All of them use the neural network based term representation. The first one, Global Transformation (**GT**), is the method without considering the local context graph. It has two variants: one is based on considering only the frequency for choosing anchors. The other one, called Global Transformation with Semantic Stability (**GT-Sem**), uses both the frequency information and semantic stability of anchor pairs (see Sec. 3.4) for solving Eq. 3.1. By comparing **GT** and **GT-Sem** we want to investigate the usefulness of setting semantic stability constraint on the anchor pairs?f selection. We also test the approach that applies the local graph (see Sec. 3.5). Since there are two variants of the global transformation, correspondingly, there will be two ways for constructing the local graphs. Thus, we test two methods, **GT-LT** and **GT-Sem+LT**, which use the global transformation and the global transformation with semantic stability, respectively.

We set the parameters as follows:

(1) num_of_dim: we experimentally set the number of dimensions of the Skip-gram model and the number of topics of LSI to 200.

(2) num_of_SFTs: we use the top 5% (18k words) of shared frequent terms to train transformation matrix. We have also experimented with other numbers but we found 5% to perform best (see Fig. 3.5 for the analysis of SFTs).

3.7.4 Experimental Results

Table 3.1: Search from [2002, 2007] to [1987, 1991]

Method	MRR	P@1	P@5	P@10	P@20
BOW	4.10E-05	0	0	0	0
LSI-Com	0.206	15.8	27.3	29.5	38.6
LSI-Tran	0.112	7.9	13.6	21.6	22.7
W2V-Com	0.225	16.5	29.7	34.1	42.9
HMM_min	0.161	13.2	20.9	20.9	24.2
HMM_avg	0.011	0	0.022	0.022	0.022
GT	0.298+	16.8	44.2+	56.8+	73.7+
GT-Sem (5,4)	0.317*+	20.2*+	45.0*+	54.0+	66.3+
GT+LT	0.369†+	24.2†+	49.5†+	63.2†+	71.6+
GT-Sem+LT	**0.387***†+	**25.3***†+	**53.8***†+	**65.9***†+	**74.7***+

Table 3.2: Search from [1987, 1991] to [2002, 2007]

Method	MRR	P@1	P@5	P@10	P@20
BOW	3.40E-05	0	0	0	0
LSI-Com	0.181	13.2	19.7	28.9	35.5
LSI-Tran	0.109	5.3	17.1	21.1	23.7
W2V-Com	0.225	15.4	29.7	38.5	43.9
HMM_min	0.171	13.4	18.6	25.8	30
HMM_avg	0.013	0.01	0.01	0.01	0.01
GT	0.226	15.2	27.3	33.3	45.5+
GT-Sem (5,4)	0.235+	15.5	29.9*	39.2*+	50.5+
GT+LT	0.235+	16.7†+	28.8	31.8	48.5†+
GT-Sem+LT	**0.258***†+	**17.5***†+	**32.0***†+	**43.3***†+	**51.5**+

Table 3.3: Search from [2002, 2007] to [1992, 1996]

Method	MRR	P@1	P@5	P@10	P@20
LSI-Com	0.115	10.6	14.9	21.3	23.4
W2V-Com	0.163	10.9	21.7	32.6	37
HMM_min	0.153	10.4	20.8	20.8	25
HMM_avg	0.012	0	0.02	0.02	0.02
GT	0.161	8.5	27.7+	40.4+	53.2+
GT-Sem (5,4)	0.172*+	6.4	33.3*+	41.7+	54.2+
GT+LT	0.202†+	10.6†	34.1†+	48.9†+	55.3+
GT-Sem+LT	**0.244***†+	**12.5***†+	**37.5***†+	**50.0**†+	**56.3**+

First, we analyze the results of finding temporal counterparts of [2002, 2007] in [1987, 1991] (Table 3.1) and in [1992, 1996] (Table 3.3). The main observation is that **GT-Sem+LT** outperforms all the baselines and other proposed methods at MRR and at the precision at different ranks. We will now discuss the results in detail.

- **Context Change Over Time.** The next observation is that the temporal counterpart finding

Table 3.4: Search from [1992, 1996] to [2002, 2007]

Method	MRR	P@1	P@5	P@10	P@20
LSI-Com	0.148	11.6	18.6	23.3	30.2
W2V-Com	0.161	8.7	21.7	32.6	37
HMM_min	0.108	9.6	11.5	15.4	19.2
HMM_avg	0.003	0	0	0	0.019
GT	0.184+	11.6+	23.3+	30.2	44.2+
GT-Sem (5,4)	0.189+	11.1+	26.7*+	35.6*+	46.7+
GT+LT	0.212†+	14.0†+	28.0†+	32.6	44.2+
GT-Sem+LT	**0.245***†+	**17.8***†+	**28.9**†+	**37.8***†+	**48.9***+

task is quite difficult. This can be seen when checking the performance of BOW which is quite poor. The correct answers in BOW approach are usually found at ranks 10k-30k (recall that the vocabulary size is 360k). This indicates little overlap in the contexts of query and counterpart terms. Since all our methods outperform all the baselines we conclude that the across-time transformation is helpful.

- **Using Neural Network Model.** When comparing the results of **LSI-Com** and **LSI-Tran** in Table 3.1 and later in Table 3.2, we can see that using the transformation does not enhance the performance when it is conducted using LSI. On the contrary, it makes the results worse. Yet, as mentioned above, applying the transformation is a good idea in the case of the Neural Network Model. We believe it is due to the difficulty in performing global transformation between topics which underlie the dimensions of LSI, in contrast to transforming "semantic dimensions" of Neural Network model.

- **Improving State-of-art.** We have found that our proposed methods statistically significantly ($p < .01$) outperform baselines **HMM_min** and **HMM_avg** in MRR and most of precision measures with paired t-test. **HMM_min** performs definitely better than **HMM_avg** since it is enough for a term to be ranked highly for only a single year-year combination. We think that **HMM_min** is more resilient for years during which a correct answer is missing from the dataset or is mentioned very infrequently. In other words, a correct term needs to be ranked highly for only a single year-to-year combination in order to be considered as a top counterpart answer. **HMM_min** has been also found to per-form best in detecting locations (MRR=0.683), when compared to persons (MRR=0.154), and objects (MRR=0.054). This is not surprising since HMM is a bag of words approach, albeit an improved one. Hence, the assumption of little change in the terms?f context is generally suitable for entities that tend to preserve con-text over time such as locations (e.g., city name change, country name change). Persons (e.g., president) or objects tend to undergo relatively large context change as time passes, hence their corresponding results are poorer than ones for locations.

- **Using Semantic Stability Constraint.** We can also observe that the constraint of semantic stability discussed in Sec. 3.4 helps to remove some frequent, yet, semantically changing words from the training set of shared frequent terms. In other words, **GT-Sem** per-forms better in picking up terms to be used as anchors for mapping two vector spaces than **GT** does. This is evidenced in Table 3.1-3.4 by **GT-Sem** which is statistically significantly outperforming **GT** and by **GT-Sem+LT** outperforming **GT+LT** (marked as *). Interestingly, we see that using the semantic stability constraint, allows **GT-Sem+LT** to achieve larger

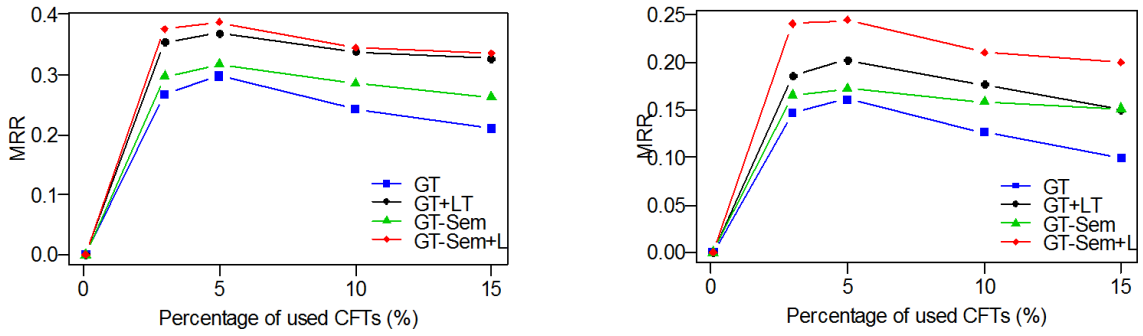


Figure 3.5: Results of MRR for GT method depending on the number of used SFTs in (a) search from [2002, 2007] to [1987, 1991]; (b) search from [2002, 2007] to [1992, 1996].

increment in MRR over GT+LT (+12.8%), than the one obtained by **GT-Sem** over **GT** (+5.0%). It can be explained that the transformation matrix trained on semantic stability constraint ensures better mapping to all the terms including the references used in local methods.

- **Using Local Context Graph.** Next, we compare the results of **GT+LT** with those of **GT** as well as those of **GT-Sem+LT** with the results of **GT-Sem** (see Table 3.1-3.4). In general, using the local context graph allows for statistically significant improvements (marked as †) over the approaches that rely only on global methods. The local method increases MRR and precision at most of cut-off values, especially, at the first rank, where the performance is boosted by 36.7%, on average. Note that GT+LT utilizes hypernyms of query as reference points in local graph. This suggests that using generalized context terms as reference points is helpful for finding correct temporal counterparts.

- **Effect of the Number of SFTs.** Fig. 3.5 shows MRR scores achieved by the proposed methods **GT**, **GT+LT**, **GT-Sem**, and **GT-Sem+LT** when using different numbers of Shared Frequent Terms (SFTs). Note that the level of 0.10% (the first point) corresponds to using 658 stop words as anchor pairs. As mentioned before, 5% of SFTs allows obtaining the best results in the case of searching from [2002, 2007] to [1987, 1991] (see Fig. 3.5(a) and searching from [2002, 2007] to [1992, 1996] (see Fig. 3.5(b)).

- **Searching from Past to Present.** We next analyze the case of searching from the past to the present in order to prove that our methods work well in both directions. This scenario applies to the case when a user (perhaps, an older person) possesses knowledge about the past term, yet, he or she does not know its modern counterpart. Another scenario envisions application in which terms in old texts are automatically "translated" for a present-day's

user. Tables 3.2 and 3.4 show the performance for searching from the past to present. We notice that **GT-Sem+LT** is the best performing method by all the measures.

- **Usefulness of Transformation.** When analyzing the performance of the baseline **W2V-Com** (see Table 3.1-3.4), we observe that **W2V-Com** has quite competitive performance compared to our proposed methods. Recall that although **W2V-Com** is also using Neural Network model, it is however not based on the concept of transformation. By analyzing the results of each test pair, we found that **W2V-Com** performs quite well for "easy" queries. To distinguish the "easy" and "difficult" queries, we categorize the query-answer pairs (1) by frequency and (2) by whether q equals tc, and test our methods based on divided test sets. The results are described below. An important thing to mention is that **W2V-Com** biases towards terms unchanged in their literal forms (i.e., the same words in both the base and target time periods). For example, for a query London the correct past counterpart is also London as the capital of England in the past (i.e., the query and its temporal counterpart have the same, unchanged literal forms). Since **LSI-Com** and **W2V-Com** are guaranteed to return London as the best result for the London query, then MRR of this query-answer pair (*London*[*present*] = *London*[*past*]) will be always 1 for both the methods. On the other hand, these methods will perform worse when a query has different syntactic form from its temporal counterpart. For example, *Obama* is the president of USA since 2009, but during [2005, 2008] he was US senator. When querying for the counterpart of *Obama* (in the sense of US president) in the target time period [2005, 2008], **LSI-Com** and **W2V-Com** would return *Obama* as the top result instead of *Bush* who was actually the president at that time. This is because, in the combined vector space, these methods cannot associate the position of *Obama* in the present with the position of *Bush* in the past, since both **LSI-Com** and **W2V-Com** lose the information of relative positions of terms within each semantic space. In the above example, the ranking of the query-answer pair (*Obama*[*present*] = *Bush*[*past*]) will never be 1 under *LSI-Com* or *W2V-Com*.

  (1) *Frequency-based evaluation.* By an "easy" query we mean here a query, which itself has high frequency in the corpus and whose temporal counterpart is also frequently appearing within the corpus. The reasoning is that during the neural network training process, a frequent query which has also a frequent counterpart is characterized by higher probability to co-occur with similar contexts within the combined document set consisting of the base time and target time documents (recall the concept of **W2V-Com** approach). To evaluate methods according to the frequency-based query difficulty, we pick up from the test set those queries that are difficult and ones that

are easy. The measure for selecting queries is based on the multiplication of query frequency and the frequency of its counterpart:

$$D(q, tc) = freq(q) \times freq(tc) \tag{3.10}$$

where q is query and tc is its correct temporal counterpart. $freq(q)$ and $freq(tc)$ are respectively the frequency of q and tc in their corresponding periods. The smaller the calculated score is, the more difficult the given query pair is.

We next rank the pairs of queries and their counterparts by the degree of difficulty in the ascending order. We then choose the top 33% percent as the difficult cases, and the bottom 33% as the easy cases. Next, we compare the performance of **W2V-Com** with the ones of the best proposed methods: **GT-Sem** and **GT-Sem+LT** for both the difficult and easy cases. According to Table 3.5, we can observe that the proposed methods **GT-Sem** and **GT-Sem+LT** significantly improve MRR measure in the case of difficult queries by on average 267% and 355%, respectively. They also increase MRR in the case of easy queries, respectively, by 23% and 39%. On the other hand, we notice that **W2V-Com** features quite poor performance for "difficult" queries, that is, queries which are infrequent and whose counterparts are also infrequent.

Table 3.5: MRR scores for difficult and easy cases by frequency. "A" is **W2V-Com**, "B" is **GT-Sem** and "C" is **GT-Sem+LT**.

Time Periods	Difficult Case (infrequent)			Easy Case (frequent)		
	A	B	C	A	B	C
[2002,2007]→[1987,1991]	0.05	**0.26**	**0.26**	0.29	0.4	**0.44**
[1987,1991]→[2002,2007]	0.02	0.14	**0.18**	0.31	0.34	**0.35**
[2002,2007]→[1992,1996]	0.05	0.07	**0.14**	0.32	0.29	**0.33**
[1992,1996]→[2002,2007]	0.1	0.07	**0.12**	0.29	0.46	**0.56**

(2) *Literal form based evaluation*. As already elucidated above, **W2V-Com** will always have a perfect mapping when q equals tc. In this sense, we regard $q = tc$, as an easy case while we consider $q \neq tc$ as a difficult case. We show the results on such divided test set in Table 3.6. Although the proposed methods lose some points in the easy case, they clearly outperform the baselines when $q \neq tc$. This confirms our reasoning described above. The percentage of test pairs where $q \neq tc$ is shown in Fig. 3.6.

Table 3.6: MRR scores for difficult and easy cases by query-answer equality. "A" is **W2V-Com**, "B" is **GT-Sem** and "C" is **GT-Sem+LT**.

Time Periods	Difficult Case (infrequent)			Easy Case (frequent)		
	A	B	C	A	B	C
[2002,2007]→[1987,1991]	0.12	0.31	**0.35**	**1**	0.72	0.67
[1987,1991]→[2002,2007]	0.10	0.16	**0.19**	**1**	0.78	0.72
[2002,2007]→[1992,1996]	0.10	0.16	**0.22**	**1**	0.42	0.67
[1992,1996]→[2002,2007]	0.12	0.14	**0.18**	**1**	0.75	**1**

Table 3.7: Examples of query answer pairs for Times Archive. (Text in italics denotes the query term in present time)

Time Periods	**US president**	**Name of location**
[2004,2009]	*Bush*	*Thailand*
[1988,1993]	Bush, Reagan	Thailand
[1967,1976]	Johnson, Nixon, Ford	Thailand
[1939,1955]	Roosevelt, Truman, Eisenhower	Siam, Thailand
[1925,1931]	Coolidge, Hoover	Siam
[1906,1915]	Roosevelt, Taft, Wilson	Siam

3.8 Experiments over Long Time Periods

In the previous section we tested our methods on the time frames ranging from 10 to 20 years. What is however unknown is the performance of the proposed approaches over longer time gaps. In such scenarios, finding correct temporal counterparts should be especially difficult due to significant change of the political and social context, technology, language, customs, etc. In addition, we expect serious problems with different word spellings and character recognition errors. Accordingly, in this section, we show the experiments on a longer dataset.

3.8.1 Datasets

To test our approach over longer and larger collection, we use the Times Archive [6]. This dataset contains 11 million digitized news articles published from 1785 to 2009 in the "The Times" newspaper. We focus on the recent 100 years' long time period (from 1906 to 2009) during which a large number of news articles have been published and the quality of the archive is still reasonably good. Users are also more likely to search within the past 100 years than in the more distant time periods. Since the Times dataset is very large (nearly 400GB size of scanned newspaper articles in total) and the data is not uniformly distributed over time, we separate the dataset by considering data amount rather than by using equal number of years as was done in the

[6] http://gale.cengage.co.uk/times.aspx/

case of the NYT corpus. In other words, we divide the timeline into time segments, which are not necessarily equal, yet, which "contain" the same size of data. In particular, we consider 20GB as a division criterium so the dataset is divided by every 20GB over time. The time periods we obtained after such division are: [2004, 2009], [1999, 2003], [1988, 1993], [1977, 1987], [1967, 1976], [1956, 1966], [1939, 1955], [1932, 1938], [1925, 1931], [1916, 1924], [1906, 1915].

3.8.2 Experimental Setup

In this experiment, our goal is to test (1) if the length of time gap will influence the performance of our methods, (2) if the proposed methods perform consistently better than the baselines, and (3) if, as well as, how much solving OCR driven problems helps to improve the results.

Since the total time period of analysis is almost 100 years?f long, we have selected 5 time periods and made the gap between the adjacent periods roughly equal to 20 years. In this way, we have chosen [1988, 1993], [1967, 1976], [1939, 1955], [1925, 1931] and [1906, 1915] as the past (target) times. The time period [2004, 2009] is always regarded as the present (base) time. In the experiments, we simulate the search from the present to each past time.

To prepare the test sets for each search task, we manually choose queries and find their correct counterparts using the same resources as when preparing the test sets for the New York Times corpus (see Sec. 3.7.2). The test sets contain locations and persons without non-entities. They are available online [7]. To make the results of every time period comparable, we issue the same query for each past period. Naturally, the temporal counterparts can be different in different time periods. Table 3.7 shows two examples of queries and their answers across different time periods. In Fig. 3.6 we display the percentage of test queries according to the equality of their literal forms: blue bars represent test pairs where query differs from its counterpart, while the red bars indicate the two are equal.

Tested Methods. For comparison, we select the best performing baseline from the first experiment on NYT, **W2V-Com**. We have chosen the sizes of SFTs to range from 2% to 5%. This is because when the gap between the base time and the target time increases, the selection of SFTs needs to be stricter. Intuitively, the language undergoes stronger evolution and thus more terms change meanings as the time gap increases.

3.8.3 Results

First, we look into the results obtained without improving OCR driven errors. Table 3.8 summarizes the results of MRR and precision @1, @5, @10, @20, @50, @100, and @500 by using all the tested methods without OCR correction. Note that unlike in the case of NYT, we decide to also check the precision at lower ranks since the Times Archive is noisier and has much larger

[7] http://www.dl.kuis.kyoto-u.ac.jp/~adam/temporalsearch_long.txt

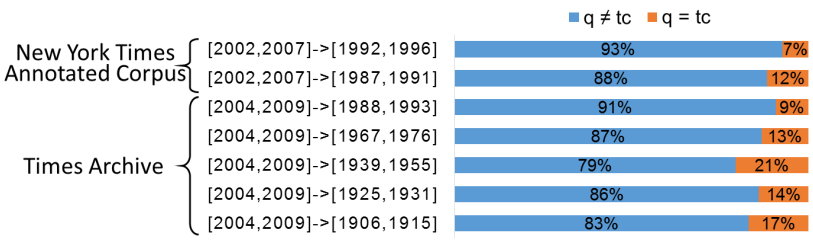


Figure 3.6: The percentage of test queries according to the equality of their literal forms: blue bars represent test pairs where query differs from its counterpart, while the red bars indicate the two are equal.

vocabulary than NYT. From the results, we can observe that **GT+LT** performs consistently better than **GT** according to MRR measure for every tested period. Also, most of the time, **GT+LT** achieves better results than **GT**. We also notice that when the time gap increases over 20 years, the performance of all the methods decreases significantly. Several reasons may cause this performance drop. One is that the decrease in the number of SFTs may impact the performance of the global transformation across time, since the smaller number of anchors could be insufficient to correctly map every dimension across vector spaces. Another reason might be due to the errors caused by OCR process. To examine the latter, we test approaches that incorporate our mapping dictionary (see Sec. 3.6) to post-process the returned list of results.

Table 3.9 shows the results obtained after improving OCR driven errors. In most of the cases we notice the improved performance (16.7% on average). Interestingly, the search tasks from the present to the distant past ([1925, 1931] and [1906, 1915]) undergo the highest improvement as measured by MRR. This makes sense as documents in the distant time periods are likely to contain more errors (e.g., due to degraded quality of paper or obsolete fonts being used) than documents from more recent time periods.

Table 3.10 shows the comparison of the results of the best performing baseline (**W2V-Com**) with the ones by the best performing proposed method (**GT+LT**). **W2V-Com** performs better than **GT+LT** in terms of MRR and precision @1, @5, @10 and @20. Recall that W2V-Com performs perfectly in the case when the query is identical to its temporal counterpart as discussed in Sec. 3.7.4. For example, when the test set pair is *Thailand*[2004, 2009] = *Thailand*[1967, 1976], **W2V-Com** is guaranteed to map *Thailand*[2004, 2009] to *Thailand*[1967, 1976] at the first rank. However, for our proposed methods, the transformation process does not guarantee the perfect mapping in such a case. This is the reason why **GT+LT** performs worse in terms of MRR as it loses for the test pairs where $q = tc$. We then expect our methods to perform better in the cases when $q \neq tc$ which is more important considering that unequal query-answer pairs are more likely to be occurred over longer time frames (or are more interesting to users). We thus

further compute MRR and precision by considering only the test pairs where $q \neq tc$. The results displayed in Table 3.11 support our reasoning that in the case of $q \neq tc$, **GT+LT** outperforms the baseline **W2V-Com** in MRR (by 28%, on average).

3.9 Summary

This work approaches the problem of finding temporal counterparts as a way to build a "bridge" across different times. Knowing corresponding terms across time can have direct usage in supporting search within longitudinal document collections or be helpful for automatically constructing evolution timelines. We first discuss the key challenge of the temporal counterpart detection in the fact that the contexts of terms significantly change over time. We then propose the global correspondence method using transformation between two vector spaces (past and present). We demonstrate two effective ways for automatically finding training sets of anchor pairs for transformation matrix. Based on global correspondence, we next introduce a more refined approach of computing the local correspondence. Finally, we propose a method for correcting OCR driven errors as a post-processing step and introduce new approach for explaining and visualizing results.

Through experiments we demonstrate that the local correspondence using global transformation with semantic stability constraint outperforms both the base-lines and the global correspondence approach. We also show that correcting OCR driven errors helps to improve MRR of the best performing method by 28%.

In the future, we plan to research the way to detect temporal counterparts from particular viewpoints or particular senses. We also plan to design methods which would make user aware that some time periods may not contain correct counterparts. One possibility for signaling the inexistence of counterpart in certain time could be done by measuring results' confidence according to a particular time span.

Table 3.8: Results of searching from present to past using Times Archive. SFT is the size of shared frequent terms chosen as anchors for training transformation matrix (without OCR errors improvement).

SFT	Time Periods	MRR	MRR	P@1 (%)		P@5 (%)		P@10 (%)		P@20 (%)		P@50 (%)		P@100 (%)		P@500 (%)	
		GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT
5%	[2004,2009]→[1988,1993]	0.135	0.163	10.3	12.4	16.5	21.6	21.6	23.7	22.7	25.8	25.8	33	35.1	40.2	55.7	60.8
5%	[2004,2009]→[1967,1976]	0.057	0.096	3.3	5.5	6.6	14.3	11	18.7	13.2	24.2	24.2	34.1	34.1	42.9	57.1	64.8
3%	[2004,2009]→[1939,1955]	0.052	0.068	0	1.9	13	14.8	18.5	18.5	21.3	22.2	27.8	26.9	33.3	32.4	53.7	56.5
2%	[2004,2009]→[1925,1931]	0.032	0.056	1.5	2.9	2.9	7.4	7.4	11.8	8.8	13.2	17.6	20.6	29.4	29.4	50	51.5
2%	[2004,2009]→[1906,1915]	0.055	0.067	3.3	3.3	6.7	11.7	10	15	13.3	15	18.3	21.7	20	25	36.7	36.7

Table 3.9: Results of searching from present to past in different time periods using Times Archive. SFT is the size of shared frequent terms chosen as anchors for training transformation matrix (with OCR errors improvement).

SFT	Time Periods	MRR	MRR	P@1 (%)		P@5 (%)		P@10 (%)		P@20 (%)		P@50 (%)		P@100 (%)		P@500 (%)	
		GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT	GT	GT+LT
5%	[2004,2009]→[1988,1993]	0.136	0.166	10.3	12.4	16.5	21.6	22.7	24.7	23.7	26.8	26.8	36.1	37.1	45.4	57.7	63.9
5%	[2004,2009]→[1967,1976]	0.061	0.102	3.3	5.5	7.7	15.4	12.1	20.9	14.3	26.4	26.4	42.9	41.8	51.6	68.1	74.7
3%	[2004,2009]→[1939,1955]	0.054	0.109	0	6.5	13.9	16.7	18.5	20.4	21.3	23.1	28.7	29.6	34.3	37	62	63
2%	[2004,2009]→[1925,1931]	0.041	0.089	1.5	5.9	4.4	11.8	8.8	16.2	10.3	20.6	25	32.4	35.3	39.7	63.2	63.2
2%	[2004,2009]→[1906,1915]	0.067	0.078	5	5	6.7	11.7	10	15	13.3	15	18.3	21.7	25	26.7	43.3	45

Table 3.10: Results of searching from present to past in different time periods using Times Archive. SFT is the size of shared frequent terms chosen as anchors for training transformation matrix (with OCR errors improvement).

Time Periods	MRR	MRR	P@1 (%)		P@5 (%)		P@10 (%)		P@20 (%)		P@50 (%)		P@100 (%)		P@500 (%)	
	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT
[2004,2009]→[1988,1993]	0.272	0.166	22.7	12.4	29.9	21.6	32	24.7	43.3	26.8	50.5	36.1	57.7	45.4	78.4	63.9
[2004,2009]→[1967,1976]	0.167	0.102	14.1	5.5	16.3	15.4	21.7	20.9	27.2	26.4	31.5	42.9	40.2	51.6	59.8	74.7
[2004,2009]→[1939,1955]	0.158	0.109	13	6.5	15.7	16.7	22.2	20.4	26.9	23.1	37	29.6	43.5	37	56.5	63
[2004,2009]→[1925,1931]	0.142	0.089	11.8	5.9	16.2	11.8	20.6	16.2	27.9	20.6	30.9	32.4	42.6	39.7	54.4	63.2
[2004,2009]→[1906,1915]	0.136	0.078	11.7	5	15	11.7	16.7	15	20	15	23.3	21.7	28.3	26.7	36.7	45

Table 3.11: Results of searching from present to past in different time periods using Times Archive. SFT is the size of shared frequent terms chosen as anchors for training transformation matrix (with OCR errors improvement).

Time Periods	MRR	MRR	P@1 (%)		P@5 (%)		P@10 (%)		P@20 (%)		P@50 (%)		P@100 (%)		P@500 (%)	
	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT	W2V	GT+LT
[2004,2009]→[1988,1993]	0.083	0.111	2.6	9.1	11.7	11.7	14.3	14.3	28.6	15.6	37.7	22.1	46.8	33.8	72.7	57.1
[2004,2009]→[1967,1976]	0.053	0.094	2.5	5.1	5.1	13.9	11.4	19	17.7	22.8	22.8	40.5	32.9	49.4	55.7	73.4
[2004,2009]→[1939,1955]	0.072	0.077	4.1	4.1	7.1	12.2	14.3	15.3	19.4	17.3	30.6	24.5	37.8	32.7	52	61.2
[2004,2009]→[1925,1931]	0.074	0.074	4.8	4.8	9.5	11.1	14.3	14.3	22.2	19	25.4	31.7	38.1	39.7	50.8	61.9
[2004,2009]→[1906,1915]	0.041	0.05	1.9	3.8	5.7	5.7	7.5	9.4	11.3	9.4	15.1	17	20.8	22.6	30.2	43.4

CHAPTER 4

DETECTING SIMILARITIES BETWEEN ENTITIES FROM DIFFERENT DOMAINS

4.1 Introduction

"The past is a foreign country: they do things differently there" is an often-quoted opening sentence of L. Hartley's novel "The Go-Between" [41]. It emphasizes common intuition that the past is quite different from the present and is generally unknown. Indeed, people tend to possess limited knowledge about things from the past. An average person typically knows mainly about events and entities taught at school. Having good knowledge and comprehension of the past is however important and useful not only for understanding the history, but also, for understanding the present and for supporting future predictions or decision making [32, 1].

Making effective use of the past often involves *across-time comparison*. Yet, for a contemporary person it is rather difficult to compare past entities with current ones. For example, it may not be easy for some to understand why *iPod* is considered similar to *Walkman*. Their similarity becomes clear after indicating that both are used to listen to music (`music - music`), both are designed to be portable (`portable - portable`) and utilize some storage media to store songs (`mp3 - cassette`) as well as both were introduced by a single, dominant company (`Apple - Sony`), etc. Likewise, it may not be straightforward to explain similar aspects of persons (e.g., *Vladimir Putin* vs. *Boris Yeltsin*) or locations (e.g., *Germany* vs. *West Germany*).

We introduce in this chapter a new problem of automatically *explaining across-time similarity* of entities to let users understand and "explore" the past through the comparison with the present. We propose an effective approach for this task, which, in a fully unsupervised fashion, returns a set of "evidences" indicating similar aspects of entities dispersed across time. Note that the task of across-time similarity explanation is not trivial. The key challenge lies in choosing the crite-

ria necessary for selecting an informative set of terms indicating similarities. Another difficulty comes from the change caused by time passage. For any pair of input entities, not only their characteristics, but also their entire context and surrounding circumstances can be quite different due to time distance, making the comparison complicated.

Our approach is corpus-based. It relies on constructing semantic vector spaces for different time periods using neural network based distributed vector representations [74], and then on aligning these representations. For any pair of entities from these time periods we discover a set of term pairs that indicate the similarity of the entities (such as the ones mentioned above for *iPod* vs. *Walkman*). We propose two methods for detecting term pairs. The first one selects terms based on three criteria: *relevance*, *semantic similarity* and *relational similarity*. The second extends the first one by graph analysis based on the concept of *systematicity* [30].

Note that to compare entities we use two document collections from two different time periods (each for either entity). This is because we cannot expect to always be able to find explicit comparative sentences that would contrast arbitrary entities, one from the present and the other from some period in the past. Note also that we do not resort to pattern detection and matching; neither, we assume the existence of any lexicon or knowledge base. Linguistic analysis of historical texts is still difficult [86], and there are no ready lexicons or knowledge bases for the past (such as Wordnet).

The methods proposed in this chapter can have diverse applications: First, they can be used for answering a special kind of questions on the comparison of past and present. Such questions cannot be easily answered by current QA systems [61, 105]. Second, they can assist historians or professionals from related fields in their work, for example, in *comparative historical studies* [38]. In particular, the proposed techniques could form components of advanced interfaces for interacting with document archives and digital libraries [116, 56]. Finally, semantic databases such as Yago [108] or DBpedia [9] could be enriched by adding novel kind of across-time links with their corresponding explanations.

To sum up, our contributions are as follows: (1) We introduce a novel problem of explaining similarity between temporally distant entities. (2) We propose two different approaches for selecting explanatory terms. (3) We evaluate the proposed methods on the New York Times Archive [104] and we demonstrate that they can successfully detect both commonalities and aligned differences of compared entities.

4.2 Background and Problem Definition

Based on the theory of *Structural Alignment Model* [30], similarity between two entities can be represented as the union of their *commonalities* and *aligned differences*.

Commonality is defined as a pair of identical features of two entities. For example, `music - music` is a commonality of *iPod* and *Walkman* suggesting that both are designed to play music.

Aligned difference is defined as a pair of features which have the same relation to both entities but have different values. For instance, `mp3 - cassette` is an aligned difference of *iPod* and *Walkman*. Although the two features are different in their literal forms, they share the similar relation of being the storage media for their entities.

Based on these definitions, we cast the problem of explaining the across-time similarity as below:

PROBLEM STATEMENT. Given two entities, e^A and e^B at different time periods T^A and T^B ($T^A \cap T^B = \varnothing$), respectively, the task is to find the set of their commonalities and aligned differences, $S_{sim}(e^A,e^B) = \{w_1^A \approx w_1^B, w_2^A \approx w_2^B, \ldots, w_l^A \approx w_l^B\}$, where w_i^A and w_i^B are terms related, respectively, to e^A and e^B.

Note that we model the task as a set construction problem, where the member of the set is a term pair denoting either commonalities or aligned differences of input entities. The constituent terms of each pair are selected from the contexts of entities.

4.3 Term Comparison across Time

Our goal is to compare entities from two disjoint time periods. However, the meaning of terms in entity's context may not be the same in the two time periods. We thus cannot directly match terms across time, i.e., by the equality of their literal forms. Even if we find the same term in both the time periods, there is no assurance if the term still denotes the same concept. On the other hand, sometimes different terms in different time periods may represent the same or very similar concept.

To compare terms, we first train word embeddings to represent the vocabularies in each time period by utilizing neural network based term representation [74, 76]. Then, we apply the transformation technique [119] to align the vocabularies of the two time periods, so that the words from one vector space can be compared to the ones in another space after transformation. The idea is to utilize semantically stable terms across time as anchors to bridge the two vector spaces. Once the mapping is found using the anchors, the other terms within the two spaces can be aligned by the similarity of their positions relative to the anchor terms in their own spaces. In this work, as anchor terms, we choose terms which have the same literal forms and which are sufficiently frequent in both the time periods (e.g., sky, river, man). One reason to choose frequent terms as anchors is because they tend to be strongly "connected" (co-occurring) with many other terms. The other reason is that frequent terms are subject to relatively small semantic drift over time. As observed in several languages including English [84, 65], the more frequent a word is, the harder

is to change its dominant meaning across time (or the longer time it takes for the meaning shift to happen).

Suppose there are k pairs of anchor terms $\{(x_1^A, x_1^B),\ldots,(x_k^A, x_k^B)\}$ where x_i^A is an anchor in one space (present) and x_i^B is its counterpart, that is, the same anchor in the other space (past). The transformation matrix $\boldsymbol{M}$ is found by minimizing the differences between $\boldsymbol{M}\cdot x_i^A$ and x_i^B (see Eq. 4.1). This is done by minimizing the sum of Euclidean 2-norms between the transformed query vectors and their counterparts. Eq. 4.1 is used for solving the regularized least squares problem ($\gamma = .02$) with regularization component used for preventing overfitting:

$$M = \underset{M}{\operatorname{argmin}} \sum_{i=1}^{k} \left\| M \cdot x_i^A - x_i^B \right\|_2^2 + \gamma \left\| M \right\|_2^2 \tag{4.1}$$

k denotes here the size of anchor terms' set containing the top 5% frequent words in the intersection of vocabularies of the two time periods. This number has been experimentally found to perform best in aligning two time periods [119].

4.4 Quality-based Similarity Detection

As explained in Sec. 4.2, the task is to select a set of term pairs denoting similarities between input entities. In this section, we first discuss the desired criteria of terms to be useful for indicating the across-time similarity of entities. Based on these criteria, we then propose several measures necessary to select high quality pairs.

4.4.1 Criteria for Selecting Term Pairs

Let $\langle w_i^A, w_i^B \rangle$ denote a pair of terms where w_i^A appears in the context of entity e^A and w_i^B occurs in the context of e^B. A high quality term pair has the following characteristics: (a) both w_i^A and w_i^B are related to their corresponding entities (i.e., to e^A and e^B, respectively); (b) w_i^A and w_i^B indicate the same or similar concept; (c) the relation between w_i^A and e^A should be similar to the one between w_i^B and e^B. Three measures correspond to the above criteria: *relevance*, *semantic similarity*, and *relational similarity*. We explain how to compute each of them in the next section.

4.4.2 Term Pair Quality Estimation

Relevance. The relevance of $\langle w_i^A, w_i^B \rangle$ to the query entities is measured using a variant of Pointwise Mutual Information (PMI). We first calculate the strength of PMI between w_i^A and e^A, and that of w_i^B and e^B. We then multiply them to estimate the relevance of the term pair to both the entities.

Semantic Similarity. We measure semantic similarity between the two context terms constituting a pair $\langle w_i^A, w_i^B \rangle$. For this, we apply transformation which was discussed in Sec. 4.3. The Semantic Similarity of $\langle w_i^A, w_i^B \rangle$ is then estimated by calculating the cosine similarity between

the transformed representation of w_i^A (i.e., $\boldsymbol{M} \cdot w_i^A$) and the representation of a term w_i^B in the other vector space, T^B.

$$S_{sim}\langle w_i^A, w_i^B \rangle = cos(\boldsymbol{M} \cdot w_i^A, w_i^B) \tag{4.2}$$

Relational Similarity. Besides the Semantic Similarity, the Relational Similarity estimates whether the relation between w_i^A and e^A corresponds to the one between w_i^B and e^B. To represent the relation, for example, between w_i^A and e^A, we take the difference of their vector representations, $w_i^A - e^A$ (see Eq. 4.3). Such linear analogical reasoning is also suggested in [76].

$$R_{sim}\langle w_i^A, w_i^B \rangle = cos(\boldsymbol{M} \cdot (w_i^A - e^A), (w_i^B - e^B)) \tag{4.3}$$

Term Pair Quality. We finally compute the quality score of a candidate term pair by aggregating the three above-discussed measures.

$$Q = Rel\langle w_i^A, w_i^B \rangle \cdot S_{sim}\langle w_i^A, w_i^B \rangle \cdot R_{sim}\langle w_i^A, w_i^B \rangle \tag{4.4}$$

4.5 Systematicity-based Similarity Detection

Sec. 4.4 introduced a direct way to measure the quality score of a term pair. Computing such score was independent of computing scores of other pairs. However, the relations (or dependences) between different term pairs can actually provide additional signal useful for choosing good explanatory pairs. Our reasoning is based on systematicity - a central factor controlling what information humans consider when comparing objects [30]. According to the *Systematicity Principle* when two compared entities have multiple features, the pair of features that preserves the maximal connected relational structure is preferred for similarity comparison. In other words, a term pair belonging to a bigger structure is preferred over isolated pair. Systematicity can be regarded as coherence reflecting term pairs' dependency on other pairs.

We propose to adopt systematicity idea to our objective and we formulate the following hypothesis:

A term pair is a good pair if it aligns well with many other good term pairs.

Alignment means here relational correspondence of term pairs. Based on the above hypothesis, we propose a graph-based approach (see Fig. 4.1). Let $G = (\Pi, E)$ be a graph composed of a set of term pairs, Π, used as vertices and the set of connecting them edges, E. The connection strength (weight) between any two nodes (term pairs), π_i ($\langle w_i^A, w_i^B \rangle$) and π_j ($\langle w_j^A, w_j^B \rangle$) is estimated based on:

- **Node-to-node relational alignment**: $align(\pi_i, \pi_j)$ - the degree to which the two nodes are relationally aligned. It measures how much the relation between w_i^A and w_j^A is similar to the relation between w_i^B and w_j^B.

- **Node pair quality**: $qual(\pi_i, \pi_j)$ - the degree to which the two nodes have high quality.

The scores of $align$ and $qual$ are measured by Eq. 4.5 and Eq. 4.6, respectively.

$$align(\pi_i, \pi_j) = cos(\boldsymbol{M}(w_i^A - w_j^A), (w_i^B - w_j^B)) \tag{4.5}$$

$$qual(\pi_i, \pi_j) = Q\langle w_i^A, w_i^B\rangle \cdot Q\langle w_j^A, w_j^B\rangle \tag{4.6}$$

The weight of an edge (ψ_{ij}) between two nodes (term pairs) is estimated as the aggregation of node-to-node relational alignment and node pair quality.

$$\psi_{ij} = align(\pi_i, \pi_j) \cdot qual(\pi_i, \pi_j) \tag{4.7}$$

Finally, we compute the systematicity score of a term pair reflecting the previously mentioned hypothesis. Specifically, scores are calculated based on the random-walk algorithm as in Eq. 4.8[1]:

$$SQ(\pi_i) = (1-d) + d \cdot \sum_{\pi_j \in N(\pi_i)} \frac{\psi_{ji}}{\sum_{\pi_k \in N(\pi_j)} \psi_{jk}} \cdot SQ(\pi_j) \tag{4.8}$$

where $N(\pi_i)$ denotes the neighbors of π_i and d is a damping factor set to 0.85. The systematicity score computation is similar to the calculation of TextRank algorithm [73]. However, in our approach each node is actually a term pair (rather than a term) and the weight of the edge depends on the alignment of term pairs incident with the edge as well as on their quality (see Fig. 4.1).

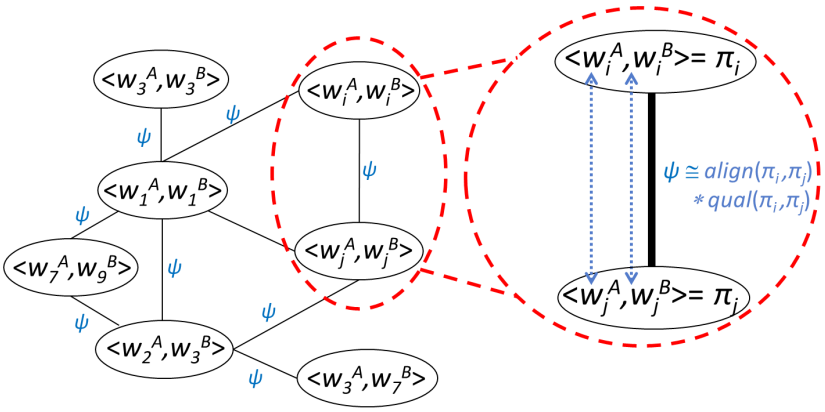


Figure 4.1: Conceptual view of graph used for systematicity-based similarity detection.

4.6 Additional Processing

4.6.1 Result Diversification

A good set of pairs is the one in which the pairs are not only of high quality but are also different from each other. In other words, pair-level diversity within the returned set should be also con-

[1]We first pruned the graph by removing nodes with low Semantic Similarity scores to make the computation manageable. The convergence criterion was set to 1e-06.

sidered. We adopt the concept of Maximal Marginal Relevance (MMR) [18] such that pairs are selected based on their quality and their dissimilarity to the already selected pairs.

$$S^* = \underset{\pi_u \in U \setminus S}{\operatorname{argmax}} \{\lambda \cdot L(\pi_u) - (1-\lambda) \max_{\pi_i \in S} sim(\pi_u, \pi_i)\} \tag{4.9}$$

U is the ranked list of term pairs retrieved by either the quality-based similarity detection method (Eq. 4.4) or by the systematicity-based one (Eq. 4.8). S is the subset of U denoting the already selected pairs. π_i and π_u denote the word pairs, $\langle w_i^A, w_i^B \rangle$ and $\langle w_u^A, w_u^B \rangle$, in the selected subset and unselected subset, respectively. $L(\pi_u)$ is the score of a term pair π_u computed by either Eq. 4.4 or by Eq. 4.8. Finally, $sim(\pi_u, \pi_i)$ denotes the semantic similarity between two term pairs calculated by multiplying the cosine similarity between w_u^A and w_i^A with the one between w_u^B and w_i^B.

4.6.2 Extraction of Supporting Sentences

We also provide supporting sentences as an additional step in order to set the detected pairs in their contexts. For each retrieved term pair $\langle w_i^A, w_i^B \rangle$, we extract two representative sentences: one containing the entity e^A and w_i^A, the other containing the entity e^B and term w_i^B. The objective of these two sentences is to implicitly address the relation between the entities and the terms constituting the selected pairs. To discover such representative sentences we first collect all the sentences (denoted as SEN^A) which contain both e^A and w_i^A and extract terms that frequently occur within these sentences to form a feature vector weighted by term frequency. Then, we compare each sentence sen ($sen \in SEN^A$) with this feature vector selecting the one with the highest cosine similarity score. The same process is applied to extract sentences containing e^B and w_i^B.

4.7 Experimental Setup

4.7.1 Datasets

For the evaluation we use the New York Times Annotated Corpus [104], which has been frequently utilized in related studies [7, 109, 119]. It contains over 1.8 million newspaper articles published from 1987 to 2007. We choose the two time periods, [2002, 2007] and [1987, 1991], which are sufficiently long and are separated by broad enough time gap. We then select query entities (e^A, e^B) such that e^A is from [2002, 2007] and e^B comes from [1987, 1991].

4.7.2 Test Sets

Since our problem is novel, there are no benchmark datasets available. We manually created test sets containing entity pairs (queries)[2]. We chose entities that are potentially similar to each

[2]Test queries' listing is attached as supplementary material.

other and which belong at equal rate to three types: objects including events (e.g., *iPod* vs. *Walkman*, *Iraq War* vs. *Gulf War*), persons (e.g., *Vladimir Putin* vs. *Boris Yeltsin*) and locations (e.g., *Germany* vs. *East Germany*). To consider diverse scenarios, half of the queries were ones involving the same entity (e.g., *Arnold Schwarzenegger* vs. *Arnold Schwarzenegger*) in both the past and present time periods, and half had different entities. In total, there were 60 different query pairs covering 90 unique entities.

In total, $4,055$ term pairs have been evaluated. We have leveraged the pooling technique [106] by pulling the top 20 retrieval results from 5 different systems (proposed methods and baselines). Three annotators judged every result (term pair) in the pool as for whether it indicates similarity of queried entities, producing, in total, $12,165$ judgments. The annotators did not know which systems generated which pairs as all the term pairs from the pool were alphabetically ordered for each query. They were encouraged to use external sources including Wikipedia and search engines in order to verify the quality of each result. The annotators could also see supportive sentences provided for every returned term pair (Sec. 4.6.2). Each annotator took on average 70 hours[3] for completing the annotation task due to the need for studying the history and searching for details of each entity. A term pair was regarded as a correct answer, if at least two annotators have accepted it. The average Fleiss' Kappa [26] is 0.71, indicating *substantial agreement* across the raters (values above 0.61 are considered as substantial agreement [64]). The average rate of commonalities to aligned differences in the ground truth is 46%:54% (43%:57%, 50%:50% and 47%:53% for objects, persons and locations, respectively, as shown in Table 4.1).

Table 4.1: Summary of test sets. **#Q** is the number of queries in each query type; **#Corr./#Pool** is the average ratio of correct answers annotated by reviewers to the average number of pooled results; **#Comm./#Corr.** and **#Al.diff./#Corr.** denote the ratio of *commonalities* and that of *aligned differences* in the correct answers, respectively.

Type	#Q	#Corr. /#Pool	#Com. /#Corr.	#Al.diff. /#Corr.	Total #Corr. /#Pool
Objects	20	30/74	13/30	17/30	607/1471
Persons	20	26/65	13/26	13/26	528/1298
Locations	20	23/64	11/23	12/23	470/1286

4.7.3 Evaluation measures and tested methods

We use $Precision$, $Recall$ and F_1 as metrics. $Precision$ is computed as the ratio of correct term pairs within the top 20 returned results. $Recall$ is calculated as the ratio of correct returned term pairs to the number of the correct pairs in ground truth.

[3]On average, more than one hour was needed to evaluate the pooled results of one query.

Table 4.2: Main results. Results marked with † are statistically significantly ($p<0.05$) better than the ones of the best-performing baseline (‡ represents significance with $p<0.01$). * indicates statistically significantly better than **QSD** ($p<0.05$).

Methods	Precision	Recall	F_1-score
Overlap	0.63	0.48	0.55
BOW	0.23	0.17	0.20
Com	0.46	0.34	0.39
QSD	0.66†	0.50†	0.57†
SSD	**0.72**‡*	**0.54**‡*	**0.61**‡*

For each tested method, we set up the same pre-processing and post-processing steps. In particular, for a given query (e^A, e^B), we extract the top relevant context terms of e^A, $\{w_1^A, ..., w_n^A\}$, and the top relevant context terms of e^B, $\{w_1^B, ..., w_n^B\}$ ($n = 500$) to be used for subsequent selection of pairs. We also apply diversification (see Sec. 4.6.1) to the results of all the methods (λ equals 0.1).

Baselines. We prepare three baselines:

(1) ***Overlap approach*** (**Overlap**): this method simply selects identical context terms of e^A and e^B. **Overlap** approach has the advantage of being simple and fast. However, it only considers commonalities between entities ignoring their aligned differences. We use it to test whether the commonalities alone are enough for the similarity comparison.

(2) ***Bag of words approach*** (**BOW**): this approach is tested to examine if the vector representation used in the proposed methods is necessary. It measures cosine similarity of any two context terms based on sentence-level co-occurrence. The pairs composed of terms most similar to each other are returned.

(3) ***Vector representation without transformation*** (**Com**): the third baseline utilizes word embeddings (Skip-gram model) same as our methods, but it does not apply transformation across time when performing similarity comparison. It merges the datasets from the two time periods and trains the word embeddings over the combined dataset.

Proposed Methods. We test the two proposed methods: **Quality-based Similarity Detection** (**QSD**) method (see Sec. 4.4) and **Systematicity-based Similarity Detection** (**SSD**) method (see Sec. 4.5) that leverages the graph-based infrastructure among the term pairs.

4.8 Experimental Results

The average scores for each method are shown in Tab. 4.2. Tab. 4.3 presents several results for example queries. The main finding is that both our methods statistically significantly outperform the baselines by all the measures. In the following subsections we discuss the results in detail.

Table 4.3: Example results. For each query we list two examples of *aligned differences* followed by two examples of *commonalities*. ✓indicates that a term pair was detected (we manually added labels shown in parentheses to indicate how terms relate to entities).

Queried entities & correct term pairs	Overlap	BOW	Com	QSD	SSD
iPod vs. ***Walkman***					
`Apple` - `Sony` (company)		✓		✓	✓
`MP3` - `cassette` (media)				✓	✓
`portable` - `portable` (characteristic)	✓			✓	✓
`music` - `music` (usage)	✓				✓
Arnold Schwarzenegger vs. ***Arnold Schwarzenegger***					
`Bustamante` - `Stallone` (competitor)				✓	✓
`Californians` - `moviegoers` (supporter)			✓	✓	✓
`Hollywood` - `Hollywood` (industry)	✓			✓	✓
`Terminator` - `Terminator` (movie)	✓		✓	✓	✓
Sepp Blatter vs. ***Joao Havenlange***					
`Klinsmann` - `Osim` (coach)				✓	✓
`Zidane` - `Vautrot` (controversy)					✓
`FIFA` - `FIFA` (organization)	✓	✓	✓	✓	✓
`soccer` - `soccer` (field)	✓	✓	✓	✓	✓
Germany vs. ***East Germany***					
`Schröder` - `Kohl` (president)				✓	✓
`Europe` - `Soviet` (union)			✓		
`Berlin` - `Berlin` (capital)	✓		✓	✓	✓
`Germans` - `Germans` (citizen)	✓		✓	✓	✓

4.8.1 Semantic Vector Representation

The next observation is that the task is quite difficult as evidenced by the poor performance of the bag of words approach (**BOW**). **BOW** measures term similarity based on the co-occurrence assumption, rather than on term semantics as in the case of the proposed methods, which utilize word embeddings. This suggests that the task of similarity explanation (e.g., *iPod* vs. *Walkman*) likely needs precise semantic mapping of context terms (e.g., a company such as `Apple` should map to another company such as `Sony`, while storage media should be connected with each other as in `MP3` - `cassette`).

4.8.2 Importance of Aligned Differences

We can observe from Table 4.2 that **Overlap** is quite competitive. Yet, it still performs worse than the proposed methods indicating that the commonality is not enough for effective similarity comparison. Since only the overlap between the context terms of two entities is considered, this method can capture only time-invariant aspects of entities. According to psychological studies [30, 66], aligned differences are actually central to the comparison process of humans and have

been found very influential in decision making. In such sense, a good system should be capable of discovering, for example, for the query: *Arnold Schwarzenegger* vs. *Arnold Schwarzenegger*, not only obvious and expected commonalities (e.g., `Hollywood` - `Hollywood`) but also the fact that *Schwarzenegger* shifted his career focus from being a movie star to serving as the governor of California (i.e., from film-making to politics). Note that both the proposed methods selected the pair `Bustamante` - `Stallone` (see Tab. 4.3) which points to this fact (`Bustamante` as a key political competitor of *Schwarzenegger* in [2002, 2007] and `Stallone` as a major rival in the film industry in [1987, 1991])[4]. However, **Overlap** cannot detect this change as it can only map a movie-oriented term from one time to the same movie-oriented term at the other time (due to its time-invariance). Moreover, even though **Overlap** performs well with commonalities for relatively short time gaps (e.g., 20 years as in the current experiments), it will likely become ineffective on longer time frames such as over 100 years, due to significant change of the world (less overlapping terms and more probability of their meaning shift).

4.8.3 Necessity of Transformation

As mentioned in Sec. 4.7.3, **Com** assigns all the terms into the same vector space without transformation. Essentially, it assumes a static world such that each term is supposed to retain its semantics over time (or has the same "position" in one common vector space based on combined documents from both the time periods). Yet, many terms tend to change their meaning and usage over time. Thus, their relative "positions" wrt. to other terms should change, too. Without the transformation, the information on relative changes of term positions in a vector space is lost. This can explain why **Com** does not select `Apple` - `Sony` pair (Table 4.3). Since the two companies existed in both the time periods, **Com** simply selects the incorrect pair: `Sony` - `Sony`, instead.

4.8.4 Commonalities vs. Aligned Differences

Table 4.4 shows the detailed performance of tested methods on detecting *commonalities* and *aligned differences*. Not surprisingly, **Overlap** outperforms other methods in commonality detection. As for *aligned differences*, the proposed methods are statistically significantly different ($p<0.01$) from the best baseline **Com** (see Al.diff. columns in Tab. 4.4), both in terms of precision and recall.

4.8.5 Usefulness of Systematicity

According to Table 4.2, **SSD** statistically significantly outperforms **QSD** across all the measures. This suggests the importance of applying the concept of systematicity for aggregating align-

[4]It is also explained by supplementary sentences in Sec. 4.8.7.

ments among candidate term pairs. **SSD** demonstrates capability of detecting term pairs which have good alignment with the entity context by considering the relation among all the candidate term pairs. This can be seen for the query *Sepp Blatter* vs. *Joao Havenlange* (FIFA presidents during [2002, 2007] and [1987, 1991], respectively) shown in Table 4.3. **SSD** selects `Zidane - Vautrot` since both the players caused controversial issues in the 2006 and 1990 World Cups, respectively, significantly impacting FIFA's reputation.

We can also see in Tab. 4.4 that **SSD** outperforms **QSD** in both commonalities and aligned differences.

Table 4.4: Results of detecting *commonalities* (Com.) and *aligned differences* (Al.diff.). * indicates results statistically significantly ($p<0.01$) better than the best performing baseline. † indicates those better than **QSD** method ($p<0.05$).

Method	Precision		Recall		F_1-score	
	Com.	Al.diff.	Com.	Al.diff.	Com.	Al.diff.
Overlap	**0.63**†	0.00	**0.99**†	0.00	**0.77**†	0.00
BOW	0.13	0.11	0.18	0.15	0.15	0.13
Com	0.25	0.22	0.37	0.30	0.30	0.25
QSD	0.29	0.37*	0.43	0.54*	0.35	0.44*
SSD	0.32†	**0.39***	0.49†	**0.57***†	0.39†	**0.47***†

4.8.6 Query Types

In Table 4.5, we compare the performance of the proposed methods over three query types. F_1 remains relatively stable for different query types, although the precision of locations is lower than the one for the other two query types. This might be due to the higher topic diversity of locations when compared to more specific objects and persons.

Table 4.5: Performance over different types of queries (Precision/Recall/F_1-score).

Method	*Objects*	*Persons*	*Locations*
QSD	.70/.47/.56	.65/.51/.56	.62/.53/.56
SSD	.75/.50/.60	.72/.56/.60	.65/.56/.62

4.8.7 Examples of Supporting Sentences

Finally, we also show two examples of supporting sentences selected for the detected term pair: `Bustamante - Stallone` (query *Arnold Schwarzenegger*(present) vs. *Arnold Schwarzenegger*(past)) suggesting similar relation (competitor or peer) between each entity and its term:

[2002, 2007]: "`Bustamante`, a democrat, is the leading candidate to replace him if the recall succeeds, holding a narrow margin over his closest competitor, *Arnold Schwarzenegger*, a republican."

[1987, 1991]: "In theatrical-release films, the big roles, and the gigantic salaries, are dominated by fellows with names like Newman, Redford, Stallone, *Schwarzenegger* and Costner."

4.9 Results Visualization

In this section, we introduce two views for result investigation consider the research topic discussed in Chapter 3 and in this chapter: the first one, called top counterpart view, visualizes the top temporal counterparts of a queried entity (returned results from Chapter 3), while the second one, similarity explanation view, displays the extracted evidences to support the understanding of across-time similarity between the query and a selected counterpart (for example, selected by using top counterpart view) (returned results from this chapter).

As discussed in Chapter 3, we map the words from the base vector space (e.g., present time) to the target vector space (e.g., past time) so that the transformed words can be then directly compared with the words in the target space. Since words are represented in relatively large number of dimensions (e.g., 200 dimensions), to visualize them we need to reduce the high dimensional space to a 2-dimensional space. We apply Principal Component Analysis (PCA) for the dimensionality reduction. Fig. 4.2 shows several examples of visualizations. The left side of Fig. 4.2 contains top counterpart views and the right side displays similarity explanation views obtained after a given counterpart candidate has been selected based on the top counterpart view. Looking, for example, at the right-hand side graph at the bottom displaying the results of *iPod* query we can see that one of its counterpart candidates, *Walkman*, has quite different semantics from the other candidates such as vcr or pc since it is located far away from them. When investigating the graph on the right hand side for the pair *iPod* and *Walkman* we notice that both share strong relation to music, both are portable and are produced by similar companies like Apple and Sony, and their storage medium is either mp3 or tape, respectively. Nano and discman seem to be other popular devices (competitors or different device types) in relation to both the input entities.

4.10 Conclusions and Future Work

The past differs greatly from the present, and is often difficult to be correctly understood. This has implications for users who wish to refer to some past entities, or for those interacting with archival document collections. In this chapter, we have introduced a novel problem of finding commonalities and aligned differences of temporally distant entities as an important step towards the objective of "bridging the past with the present". We have proposed two unsupervised methods to solve this task and have successfully demonstrated their effectiveness.

In the future we plan to extend our approach to selecting also non-aligned differences. In

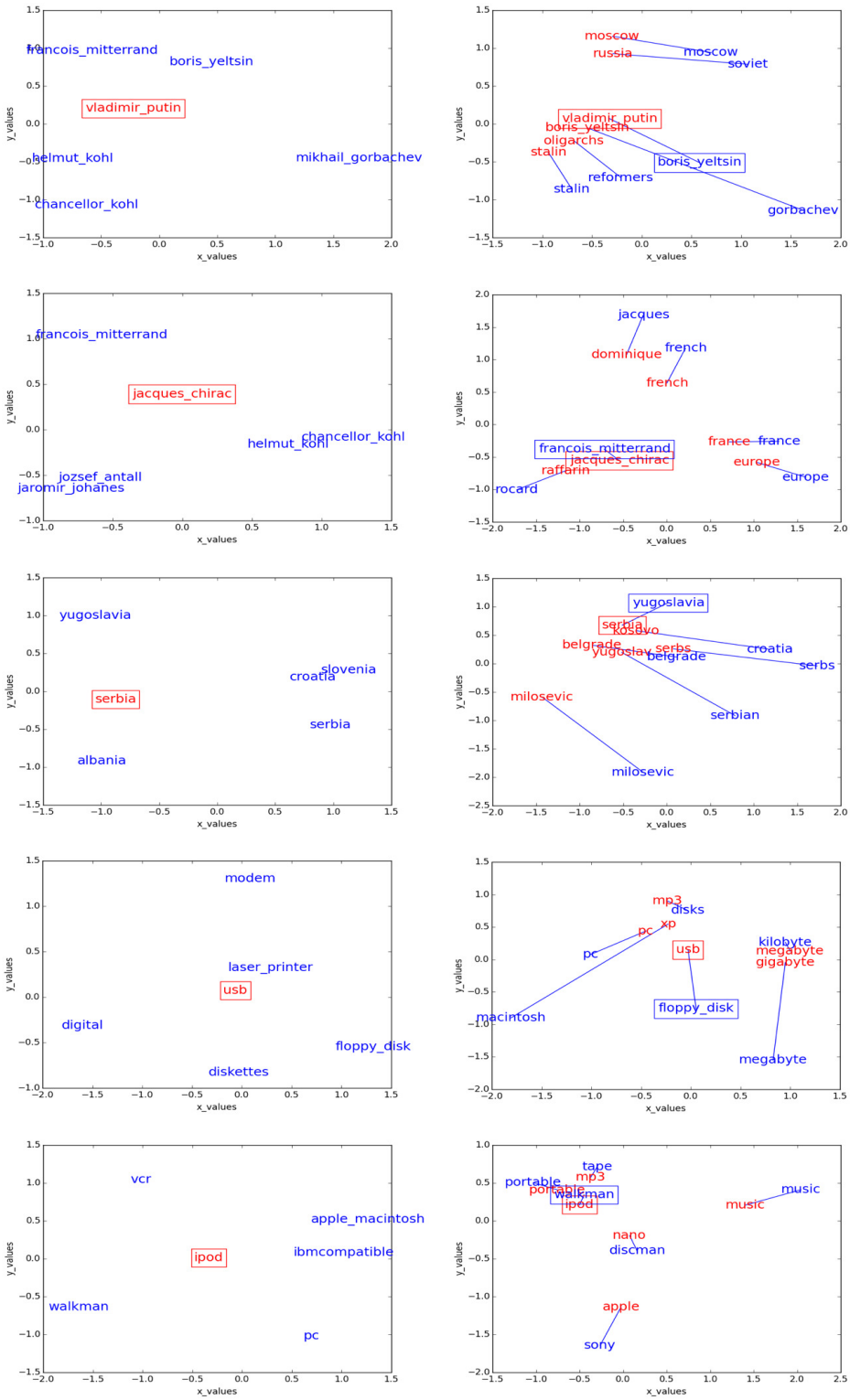


Figure 4.2: Examples of two views for result presentation from [2002,2007] to [1987,1991] (graphs on the left show top counterpart views containing top 5 candidates for a given query (depicted as a framed term); the graphs on the right are similarity explanation views showing top 5 evidences as interconnected term pairs; red-colored terms are from present while blue-colored ones are from past.

addition, we will test our methods on entities from more distant time periods (e.g., over 100 years), which necessarily requires using special datasets and, likely, the correction of OCR-driven errors.

CHAPTER 5

DETECTING CAUSE-EFFECT RELATIONSHIPS IN TEXT ARCHIVE

5.1 Introduction

Typically, within the data mining research, product reviews were analyzed for determining positive or negative aspects of products and for summarizing user opinions about them [25, 43, 44, 68, 87, 110]. However, given the existence of review archives containing reviews written over spans of many years, it has become now possible to undertake longitudinal studies to tackle evolutionary aspects of products. In this context, an interesting aspect of product reviews is their social character. They are written by users who usually either own or, at least, have used the reviewed products; hence, the reviews contain hints on the way in which products were used including the activities related to the products, the places and situations when they were used and so on. Based on this observation, we propose to use such temporal collections of product reviews for analyzing how the evolution of products influences users and their lives over time. For example, it is known that early music devices equipped with batteries and earphones like Walkman were small and efficient enough to be used for listening to music while performing outdoor sports and activities. One can then say that product features (e.g. earphones and batteries) had social impact on user lifestyle (e.g. performing outdoor sports). In this work we aim to automatically find such relationships by analyzing longitudinal archives of product reviews.

Our approach is quite novel. Although, the research of evolution has been carried in domains such as biology and sociology, efforts towards computer-assisted evolution analysis are quite sparse. Within the computer science, researchers approached the evolution of topics [3, 22, 118], named entities [70, 109] and terminology [7, 48, 49, 62, 119]. Yet, few works attempted the task of automatically analyzing and portraying the evolution of products. To the best of our

knowledge, no prior work proposed computationally determining changes in social lives triggered by the technology evolution.

There are many potential benefits of automatic approaches towards detecting the influence that products and technology had on our society. First, scientists in the areas of history of science/technology as well as areas of social studies can be assisted in their research, and, in particular, in verification of new hypotheses or exploration of novel ideas. Given any quantitative data derived from reviews they can be better informed about the actual progress of technology and its relation to social evolution. Actually, the phenomenon of technology impact on society has been an interest of social scientists and historians since long ago. Within this area, the concept of technological determinism [20] presumes that the technology drives the development of social structure and cultural values. However, as far as we know, no data mining solutions have been specifically offered for studying this phenomenon. Apart from scientists, average users may be also interested in learning more about the products they use. They could then appreciate the complexity and progress of engineering and technology. One can imagine online applications that upon a user-issued query could construct visual overviews of how our society created and used products in the past. Ideally, various types of queries should be allowed in such systems ranging from particular product models (e.g., iPhone) to the entire categories of products (e.g., music devices). We believe there are multiple educations opportunities in data-driven visualization and interactive exploration of evolution in which social impact of products would be particularly emphasized. Finally, producers should be interested in seeing how the models released in the past were accepted and how they were used by the society. Besides focusing on particular product lines they should be also interested in general findings such as any regularities or, rather, any exceptional cases of the technology progress and impact on society. The possible applications here could involve better product design and supporting informed discussion on the relation between products and users.

Ideally, the statistical analysis on product-related datasets should not only explain the evolution of a single product model, but should also allow generalizing to entire product categories. In other words, it should be possible to apply a generic evolution analysis model over different scopes (or categories) to be able to analyze the technology evolution within wider scopes. For example, rather than analyzing the study of the evolution of Walkman (a particular product), the evolution of portable music devices (a product category) and, then, music devices in general (a coarser product category) should be also made possible. Hence, the proposed solution need to be flexible enough to allow for coping with different granularity levels and hierarchies of data.

In this chapter we approach the task of evolution analysis according to the requirements discussed above. To extract the evidences of the technology impact on human activities we search

for causal implications in which one term has an influence relationship on another term. Our assumption underlying the causality relationship is that changes of certain terms trigger changes of other terms. Note that this approach is substantially different from the typical causal relationship detection methods studied within NLP area [10, 19, 35, 34, 42, 51, 53, 78, 88]. Rather than parsing direct evidences of causality or entailment expressed in natural language, we search for actual evidences of such relationships within temporal document collections. This allows for a generic method that does not require explicit evidences such as actual descriptions of product impact. These will be implicitly derived and inferred by analyzing time series related to terms within the underlying collection.

Our approach is as follows. We detect changes related to selected terms over time by two methods: tracking fluctuations in term popularity and tracking fluctuations in the way terms are used. The former relies on simple frequency measuring over time, while the former captures contextual changes related to given terms. To detect contextual changes we sequentially retrain neural network based term representations on temporal subsets of underlying document collection.

Having detected terms with either type of a change, we search for any terms whose change could be affected by another terms change. For this we employ temporal logic approach [39, 54, 60]. Once causal term pairs are selected, the next step is to group them in order to find concept-to-concept causality scenarios in which both the cause and the effect are represented by sets of terms rather than by single terms. The concept-level causality is obtained by aggregating binary causal relationships. We propose several approaches for such causal pair aggregation. Note that in some scenarios more than one technology can trigger given changes in product usage. Thus one of the proposed aggregation approaches allows also for grouping semantically unrelated terms in order to detect the so-called co-causality scenarios.

Analyzing the influence of each term on every other term is impractical or, at least, computationally expensive. Thus, to decrease the number of candidate pairs we distinguish two kinds of product features: physical features and conceptual features. The former denote the product specifications or features describing product components and outlook. On the other hand, conceptual terms refer to the way in which products are used by consumers including any activities enabled by the products, places where products are used and more general circumstances of products usage. We select the physical features by classification approach, while the conceptual features are represented as verbs or location names.

In this chapter we also experiment with one more method that can detect causality by simulating alternative history. In order to investigate the change brought by a given feature or a product model we temporarily remove any related content from the entire dataset. We then train

neural network on such a modified dataset to compare the new word representations with the corresponding representations based on the original dataset. This allows finding words subject to considerable context change due to the removal of data related to a given feature or model. For experimentation, we use the Amazon Review Dataset [71], which contains over 34 million of reviews about more than 2.4 million products written by 6.6 million users from June 1995 to March 2013. We conduct experiments on three large categories of products: Electronics, Portable Audio & Video, Electronics, Camera & Photo and Electronics, Computers & Accessories, Laptops and we demonstrate that our approach can return information better and more useful than the one delivered by baselines.

The main contributions of this chapter can be summarized as follows:

1. We propose a novel concept of analyzing product evolution by considering its social influence over time.

2. We introduce a novel approach for implicit causal relation extraction from temporal document collections. The proposed methods can be flexibly applied over different levels of product categories as well as they are generic enough to be utilized in other scenarios besides the studies of the technology evolution.

3. We evaluate our approach by experiments on the Amazon Review Dataset that spans 18 years.

The remainder of this chapter is organized as follows. Section 5.2 gives background of main approaches on causality detection and our formal problem statement. In Section 4 we outline the proposed method for constructing term-related time series based on review datasets. Section 5 contains methodology for detecting causal relations based on changes related to words. Section 6 describes additional method for detecting causality that relies on data removal of collection subsets from dataset. We report experimental results and provide discussions in the next section. We provide additional discussions in Section 8. Finally, Section 9 concludes the chapter and outlines out future work.

5.2 Problem Definition

We define here key concepts related to the proposed approach.

Definition 1. *Product category* is set of products that share common features or objectives.

The input data for our methods is any category level of products. For example, portable music devices or laptops are product categories; same as, music devices and computers which are more general categories.

Definition 2. *Physical features* are product attributes that can be named (and often perceived) and that are integral parts of a product.

In this work, physical features mainly represent products components and specifications. To detect physical features we employ a dedicated classification approach (described in Sec. 5.6.2).

Definition 3. *Conceptual features* are actions, situations or social behaviors that occur thanks to the usage of products.

In this work, for simplicity, we regard verbs (e.g., navigate, scroll) and situation terms (e.g., gym, home) as conceptual features. Note that other words can be used here depending on specific needs or applications.

Definition 4. *Causal relationship*, whose strength is denoted by I(c,e), is the relation between a cause term c and an effect term e, where the change in the cause leads to the change in the effect. Informally, the causal relationship is conceptually portrayed in Fig. 5.1.

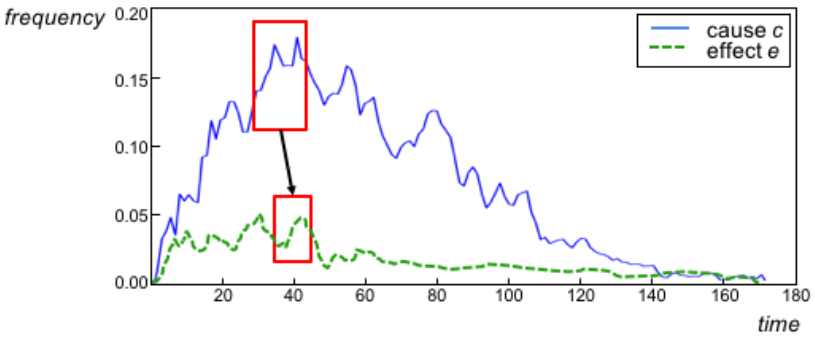


Figure 5.1: Conceptual view of a causal relationship. The black arrow represents the direction of causal relationship.

In this work, we assume that a cause is the change occurring in one of physical features and the effect is the change occurring in one of conceptual words. We then define $I(c, e)$ as the combination of functions:

$$I(c, e) = f(d(c), d(e)) \tag{5.1}$$

Here d is a function (described in Sec. 5.4.1) which quantifies the change of a cause term c and the change of an effect term e. The function f (described in Sec. 5.4.2) estimates causal strength between the two types of changes. The output of our methods consists of the ranked list of detected causal relations occurring between physical features and conceptual features in each unit time (e.g., a month or a year). Fig. 5.2 shows a high-level overview of our proposed approach, which contains three fundamental stages: pre-processing, change detection and causality detection. We will describe each part in the following sections.

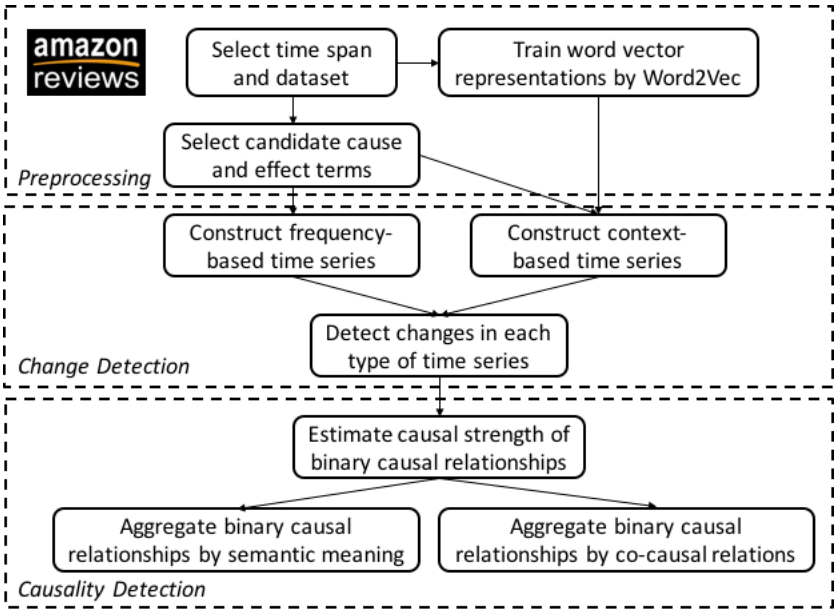


Figure 5.2: High-level overview of the proposed approach.

5.3 Representing Term Change over Time

As mentioned before, we first need to detect changes related to terms over time based on the assumption that only the terms subject to considerable temporal changes can constitute causes or effects. Hence, constructing time series is the first step for choosing candidate terms for detecting causal relations. In this section, we introduce two approaches for measuring changes, the frequency-based and context-based change detection. The former tracks the fluctuations in term occurrence over time, while the latter measures variations in words context and, thus, the way in which the term is used.

5.3.1 Term Occurrence

When analyzing diachronic corpora, one commonly measures term frequency in different time periods. By comparing the number of times terms appear over time we can then detect periods of high and low usage of terms. In the case of physical features in our dataset (product reviews) the periods of increased frequency denote time when new innovations appeared or new kind of product characteristics become common. On the other hand, the decreasing term usage indicates either the disappearance of a feature or the fact that it is no longer special or interesting to be mentioned. For measuring the term frequency, we split the dataset into equal length, non-overlapping time units. Then the average frequency of a term per document is computed at each time unit as follows:

$$Freq_t(w) = \frac{|D_t|w \in D_t|}{|D_t|} \tag{5.2}$$

$\frac{|D_t|w \in D_t|}{|D_t|}$ is the number of times a word w appears in the document set D_t within a unit slot t. $|D_t|$ is the number of documents created at t. The advantages of the frequency-based method are its ease of implementation and scalability over large datasets. On the flip side, it cannot capture more subtle changes, such as ones related to term usage.

5.3.2 Term Context

As mentioned before, simple term counting cannot distinguish contextual changes in cases when a word is used with relatively stable frequency, yet, its usage and surrounding context change. For example, given the category of portable music devices, the occurrence count of a term car, which is considered a conceptual term describing a situation/place (i.e., a place where users listen to music in this case), does not change much across time. On the other hand, its context changes according to the types of music devices used while travelling by car (i.e., devices ranging from cassette-based ones, through CD-based to mp3-based ones).

Context shifts can be detected by analyzing fluctuations in the distributed representation of words [40]. The distributed representation of words is based on the hypothesis that words appearing in similar contexts are semantically similar. It enables to measure the semantic similarity of words as the distance between their vectors. The distributed representation of words can be computed by applying neural networks, as first proposed by Rumelhart et al. [102]. Later, Mikolov et al. [74, 76] proposed a novel way to overcome efficiency issues in computing distributed representation of words. Their model, called Skip-gram, utilizes a simplified neural network architecture for learning vector representations. It has the following merits: (1) ability to capture precise semantic word relationships; (2) scalability to millions of words.

Fig. 5.3 shows the overview of the process of constructing context-based time series of words. In our experiments, we use Genism implementation of Skip-gram model in Python 2.7 programming language. We set the length of the context window size to 10. As for the size of dimensions of the word vector presentations, it is natural that the dimension count should be bound to the vocabulary size, as also stated in Mikolov et al. [74]. The more dimensions are used, the more precisely we are able to represent the semantic meaning of words (Mikolov et al, [74]). However, this comes with the price of an increased training time. Besides, the generalization capability may be impaired if the number of dimensions is set to the too high level. Currently, we use 100 dimensions for the experiments on the Amazon Product Review Data.

First, we construct a vocabulary list by collecting all the words that occurred more than certain threshold (5 times) at any time within our dataset. Then, based on such combined vocabulary we

train the Skip-gram model using the reviews published only in the first year. Thus from the beginning, every word is going to have a position in the vector space. Note that, for those terms which have not appeared in the first years, the model still assigns some initial vectors. Next, we sequentially retrain the initial vectors using the reviews published for each subsequent month (i.e., we iterate over epochs and train the word vectors until convergence). The convergence is defined as the average angular change in word vectors between epochs:

$$\rho = \frac{1}{|V_t|} \sum_{w \in V_t} arccos \frac{\chi_w(t, \varepsilon) \cdot \chi_w(t, \varepsilon - 1)}{\|\chi_w(t, \varepsilon)\|_2 \, \|\chi_w(t, \varepsilon - 1)\|_2} \tag{5.3}$$

where the $\chi_w(t, \omega)$ is the vector of word w at time slot t and epoch ω. The model will stop updating the word vectors when ρ is lower than 0.0001.

Finally, we construct the time series of a word w by computing the distance between its distributional representation at time t and at $t - 1$ (see Fig. 5.3). The distance is measured as follows:

$$Dist_t(w) = 1 - \frac{\chi_w(t) \cdot \chi_w(t - 1)}{\|\chi_w(t)\|_2 \, \|\chi_w(t - 1)\|_2} \tag{5.4}$$

Note that our process of constructing contextual change is different from the one proposed by Kulkarni et al. [62]. Their approach assumes that if a term is stable in its meaning then its neighbors in the vector space would be also retained. Therefore, for each term, they compute k-nearest neighbors and enforce the position of the neighbor terms to be the same across adjacent time periods by applying transformation techniques. If a term w in time t cannot be perfectly mapped to itself in $t + 1$, it means it underwent a contextual change. Their approach only tracks the context change of terms that existed from the beginning (e.g., gay). Newly appearing terms (e.g., USB, mp3) are ignored from the beginning unlike in our approach. Also, since our approach is based on retraining, we save the computational time spent during transformation period.

We also note that the method we use for capturing the context-based time series is based on the sequential retraining of terms' vector representations. This means that for a given time unit t we utilize the trained vectors from the previous time unit $t - 1$ in such a way that we only adjust these vectors based on the newly observed data from the time unit t. Hence, if there is no data at t, the vector representation of terms will remain the same. However, under the closed world assumption, we are aware that we may not capture the actual changes that terms underwent across time in case of sparse data.

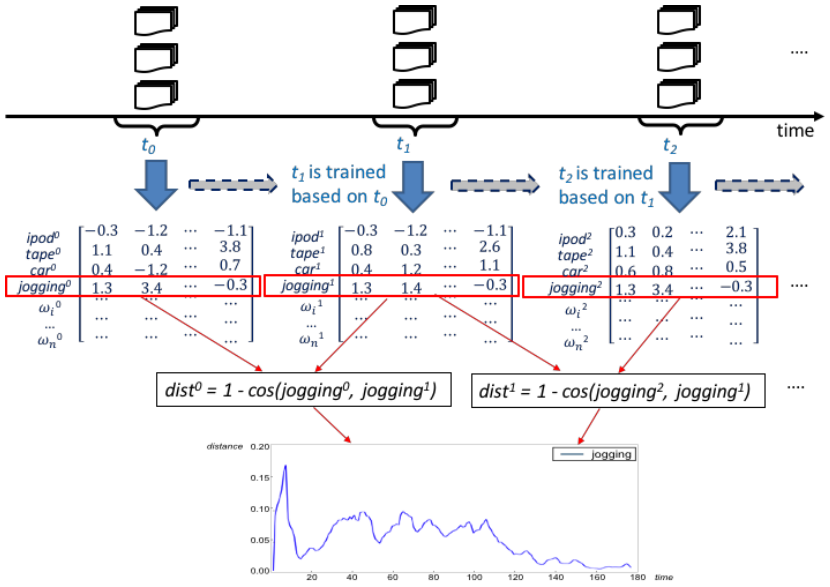


Figure 5.3: Constructing context-based time series.

5.4 Detecting Causal Relations

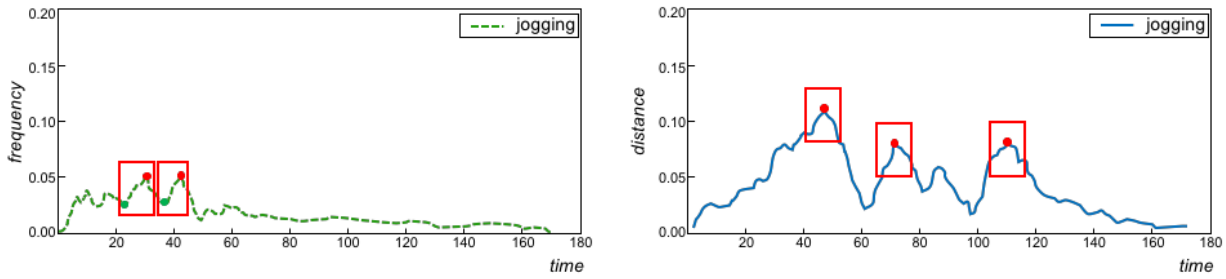
In Sec. 5.2, we explained that only terms undergoing changes can take part in causal relations. By constraining the cause and effect to be related to only the changing terms (either frequency-based or context-based changes) we effectively decrease the number of candidate pairs to be tested. According to the discussion in Sec. 5.2, we explain below the first component of causality computation - function d used for detecting the changes in time series. Next, we describe the composition function f by first estimating the causality strength between every pair of a physical-term change and a conceptual-term change and then by aggregating binary causal relationships.

5.4.1 Detecting Term Change

Changes in the two above-discussed time series representations of words have different characteristics. The increase in word frequency means that the word becomes more popular. This could be due to either some change in existing technology, introduction of a novel technology or the change of concepts related/dependent on the technology.

On the other hand, for the context-based time series, each data point represents the dissimilarity (distance) obtained by comparing word context at a given time point to the one at the previous time point. Thus, the peak points (or points above certain threshold) mean a high context change, while points with low distance denote time periods with little or the lack of any context changes when compared to the previous time points. In this work, we assume that the peak periods represent the change periods of term's context. We thus detect periods of high change in the frequency-based time series and peaks in the context-based time series.

To detect the time periods of high increases in frequency, we assume that the increase period is the time frame between the adjacent valley and peak. We use the method by Billauer et al. [8] for detecting peaks and valleys. A peak is considered as the highest point between valleys. This method uses the distance (denoted as lookahead) to look ahead from a peak candidate. A minimum difference (denoted as delta) between a peak and the following points is also specified to distinguish an actual peak from noise. After detecting the peaks, we further estimate the change period by considering the parameter slope, which guarantees the absolute value of a minimum slope within the change period. In Alg. 1 we describe the process of the change detection within both the frequency-based time series (part 1 and 2) and the context-based time series (part 1). Note that other approaches for change detection in time series such as [55] could be applied here. For every term (cause and effect terms), we apply the method described above to detect the change periods in its frequency-based time series and context-based time series, respectively. Fig. 5.4 shows an example of change periods detected for the term jogging in its frequency-based time series (Fig. 5.4a) and context-based time series (Fig. 5.4b).



(a) the detected change period as the increase period within the frequency change.

(b) the detected change periods as the peak periods of the context change.

Figure 5.4: Example of change periods detected for term jogging. Small green diamonds indicate the valleys of the time series, and the red diamonds represent the peaks of the time series. Red rectangles mark the detected change periods.

5.4.2 Detecting Causality

Before estimating the causal strength of a potential cause on a potential effect, we first determine the time period during which the causal relations occur. Based on the change periods of the cause and the effect (as detected in Sec. 5.4.1, the causal strength can be computed between a pair of candidate cause and effect only if their change periods satisfy the following requirements:

(1) the change period of the cause term appears before the one of the effect: $Start_{cause} < Start_{effect}$;

(2) the change period of the cause and the change period of the effect term have non-zero overlap

Algorithm 2 Change Detection in Time Series

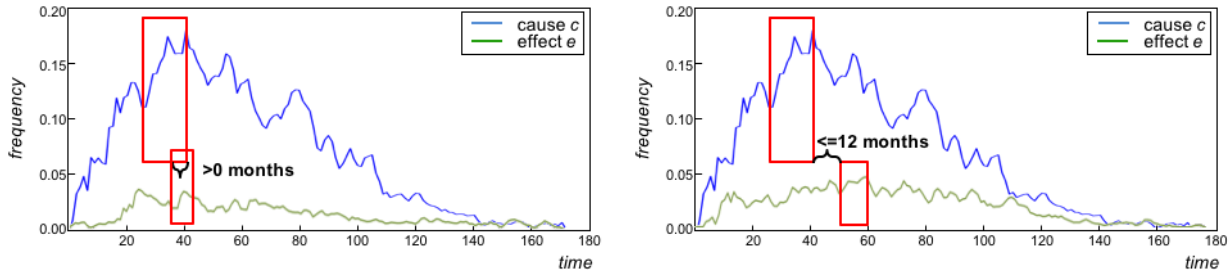
Input: Time series $\mathcal{T}$, Parameters $lookahead$, $delta$, $slope$.
Output: Change periods ($Peak_{\mathcal{T}}$ or $Increase_{\mathcal{T}}$).

/* Part 1: Peak and Valley Detection */
$minima, maxima \leftarrow \infty, -\infty$
for $\mathcal{T}_i \in \mathcal{T}$ **do**
  if $\mathcal{T}_i > maxima$ **then**
    $maxima \leftarrow \mathcal{T}_i$
  end if
  if $\mathcal{T}_i < minima$ **then**
    $minima \leftarrow \mathcal{T}_i$
  end if
  if $\mathcal{T}_i < maxima - delta$ and $maxima \neq \infty$ **then**
    if $maxima > \{\mathcal{T}_k | \mathcal{T}_k \in [\mathcal{T}_i, \mathcal{T}_{i+lookahead}]$ **then**
      $Peak_{\mathcal{T}} \leftarrow$ add($\mathcal{T}_i, maxima$)
      $maxima, minima \leftarrow \infty, \infty$
    end if
  end if
  if $\mathcal{T}_i > minima + delta$ and $minima \neq -\infty$ **then**
    if $minima < \{\mathcal{T}_k | \mathcal{T}_k \in [\mathcal{T}_i, \mathcal{T}_{i+lookahead}]$ **then**
      $Valley_{\mathcal{T}} \leftarrow$ add($\mathcal{T}_i, minima$)
      $maxima, minima \leftarrow -\infty, -\infty$
    end if
  end if
end for
/* Part 2: Increase Detection */
for adjacent($valley, peak$) $\in (Valley, Peak)$ **do**
  $leftbound \leftarrow (\mathcal{T}_{peak} - \mathcal{T}_{leftbound})/(peak - leftbound) > slope$
  $Increase_{\mathcal{T}} \leftarrow$ add($leftbound, peak$)
end for

(see Fig. 5.5a or are separated by a lag shorter than the maximum allowed lag (see Fig. 5.5b: Overlap([$Start_{cause}, End_{cause}$], [$Start_{effect}, End_{effect}$]) > 0 or ($Start_{effect} End_{cause}$) $\leq MaxLag$ (by default, $MaxLag$ is equal to 12 months).

In other words, we assume the lack of any causal relation if the change period of the cause term and the one of the effect term have no overlap with each other, and are separated by time longer than $MaxLag$. Furthermore, no causal relation will be detected in the case when the effect terms change period appears before the one of the cause term. The impact time $[t_s, t_e]$ is defined as $[Start_{cause}, End_{effect}]$.

As mentioned before, we adapt temporal logic approach proposed by Kleinberg et al. [59, 58] to measure the causality strength between the candidate pairs of a cause c and an effect e. Our



(a) Case when the change periods of cause and effect terms have non-zero overlap.

(b) Case when the change periods of cause and effect terms are not farther than $MaxLag$.

Figure 5.5: Conditions for the change periods of terms required to consider terms as candidates for causal relationship.

approach assumes the key principles of probabilistic causality that:

1. c temporally precedes e;

2. the occurrence of c raises the probability of predicting e.

Based on the above principles, we formulate the measure of a causal strength in Eq. 5.5. Intuitively, the strength with which c causes e depends on how much the probability of the occurrence of e is increased given the occurrence of c. Since we deal with the token-level causality, each causal relation occurs in a certain causal time period $[t_s, t_e]$. Thus the causal relation between the same words may be characterized by different causal strengths at different time periods.

$$\begin{aligned} I_{[t_s,t_e]}(c,e) &= P_{[t_s,t_e]}(e|c) - P_{[t_s,t_e]}(e|\neg c) \\ &= \frac{tf(e \in M_{[t_s,t_e]}(c))}{\sum_{i \in V} tf(i \in M_{[t_s,t_e]}(c))} - \frac{tf(e \in M_{[t_s,t_e]}(\neg c))}{\sum_{i \in V} tf(i \in M_{[t_s,t_e]}(\neg c))} \end{aligned} \tag{5.5}$$

$P[t_s, t_e](e|c)$ denotes the probability of seeing an effect e given the occurrence of cause c within $[t_s, t_e]$. It is computed by dividing the effect e's frequency in the documents which contain the cause c by the corresponding frequency of e in the documents which lack c. $M[t_s, t_e](c)$ denotes the set of documents which contain c at $[t_s, t_e]$, and $M[t_s, t_e](\neg c)$ is the set of documents without c during the same time period.

Note that computing causal strengths using Eq. 5.5 has disadvantage that causal relations might depend on other potential causes of the same effect. In reality, several causes may cause the same effect, or there can be an additional hidden cause that implies both c and e. Lastly, c may not be a genuine cause being a spurious or just a weak cause.

We then improve Eq. 5.5 to more accurately detect actual causes. In particular, we determine whether a given term c is a significant cause of e by considering all candidate causes of e denoted

by X, based on the list of initial candidate causes computed using Eq. 5.5. The idea is to calculate the average difference in the probability of an effect for each of its candidate causes, in relation to all other candidate causes of the same effect. The target cause is significant if the average probability of effect occurrence substantially differs depending on when the target cause is present or absent. Such global causal strength of $c \rightarrow e$ is thus computed by applying Eqs. 5.6 and 5.7.

$$I_{global}^{[t_s,t_e]}(c,e) = \frac{\sum\limits_{x \in X \setminus c} \varepsilon_x^{[t_s,t_e]}(c,e)}{|X \setminus c|} \tag{5.6}$$

where,

$$\begin{aligned} \varepsilon_x^{[t_s,t_e]}(c,e) &= P_{[t_s,t_e]}(e|c \wedge x) - P_{[t_s,t_e]}(e|\neg c \wedge x) \\ &= \frac{tf(e \in M_{[t_s,t_e]}(c \wedge x))}{\sum\limits_{i \in V} tf(i \in M_{[t_s,t_e]}(c \wedge x))} - \frac{tf(e \in M_{[t_s,t_e]}(\neg c \wedge x))}{\sum\limits_{i \in V} tf(i \in M_{[t_s,t_e]}(\neg c \wedge x))} \end{aligned} \tag{5.7}$$

Eqs. 5.6 and 5.7 estimate the causal power of a given candidate cause c by measuring how much on average the co-occurrence of c with other candidate causes can explain the fact that e occurred. denotes here the change of the probability of the effect when removing a target cause c. The larger the , the more significantly the target cause c impacts the effect. Note that since we use the set X of pre-selected candidates by applying Eq. 5.5 we can keep the computation time of Eqs. 5.6 and 5.7 on manageable level.

Note that to compute the causal strength, we have assumed a probabilistic approach. This naturally involves the computation of term co-occurrence within the dataset. However, unlike the standard calculation of the co-occurrence or correlation, we consider the dynamics of term appearance over time. In particular, we ensure the first rule of causality, that is, the cause must precede the effect. Next, our unit of concern is not a term itself, but the change related to the term such as the change of its frequency or the change of term's context over. This means that the causality computation is bound to the detection of term change, so that a given "change" causes some other "change" within the dataset. In particular, even if two terms have high correlation within the time window of fixed length, our method will not consider them to be in a causal relationship unless both of these terms undergo substantial change in that time window (two requirements listed above). Finally, according to the two key principles of probabilistic causality (two key principles above), our approach compares two probabilities: the probability of an effect (i.e., some change) under the occurrence of the cause (another change), with the corresponding probability of this effect without the causes occurrence. In other words, this criteria involves checking if the occurrence of a cause does actually increases the probability of the occurrence of the effect.

5.4.3 Aggregating Binary Causal Relations by Meaning

In this section we describe further extensions to the causality computation. Instead of binary causalities involving pairs of single terms as approached until now, we propose here combining similar or duplicate causal pairs. For example, the following three binary relationships essentially mean similar things: `mp3`→`walking`, `minidisc`→`running`, `minidisc`→`jogging`. All of them indicate that activities similar to jogging can be accompanied by an additional activity of listening to music thanks to novel technologies such as `mp3` and `minidisc`. Thus, the above relationships could be combined. In this case, the above mentioned binary causal relations can be grouped as {`mp3`, `minidisc`}→{`walking`, `running`, `jogging`}. Note that we use angular braces to represent a group of words with similar meaning.

The objective of this aggregation type is aggregating semantically similar causal relations. The first approach, which we will discuss below, is based on independent aggregation of both causes and effects to form semantic groups on each side. The second approach relies on the simultaneous aggregation of the both sides.

Grouping by Similar Concepts. We show in Fig. 5.6 the overview of the first aggregation process. The figure is composed of two parts. The left hand side shows the original graph composed of nodes denoting cause and effect terms. Here, the thin, gray, undirected links indicate semantic similarities between the nodes. The red directed links denote the discovered cause-effect relations (Eqs. 5.6 and 5.7). The grouping procedure is shown in the right hand side graph. After performing the aggregation (Step 1) some nodes related to the cause terms get combined into clusters as shown in the right hand side graph. The same happens with the nodes that indicate the effect terms. Finally, in the Step 2, the scores of the causal strengths are computed between all the clusters on both the sides.

Below, we will describe in detail Step 1 and Step 2 of the grouping procedure. We also summarize the entire aggregation algorithm in Algorithm 2.

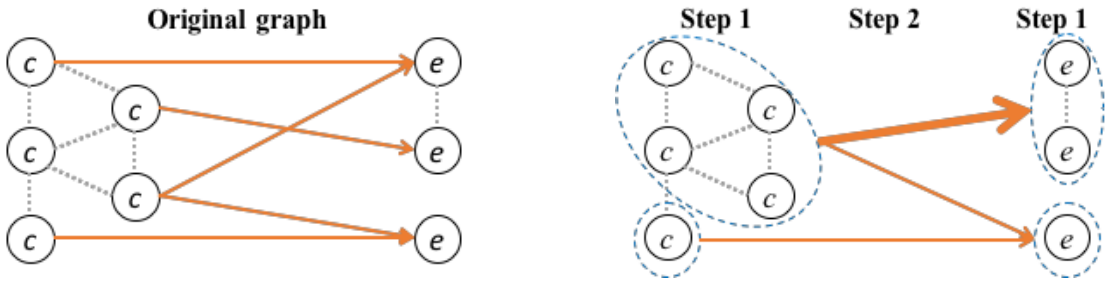


Figure 5.6: Aggregation method by similar concepts.

Step 1. We first apply a simple clustering process in order to separately discover semantic

groups on the cause and effect sides. Take the cause side as an example. We first construct a similarity graph for the causes. In such a graph any two causes will be connected if their semantic similarity is above given threshold (the threshold is equal to 0.5, by default). Next, we sort all the pairs of connected causes by the value of their semantic similarity. The clustering process proceeds then as follows. We start from the pair having the highest similarity (c_i, c_j) and we find the common neighbors of c_i and c_j. c_i and c_j are then grouped with such common neighbors. Next, the second pair (i.e., the pair of causes that have the second highest similarity value) is taken and the processing repeats in the same way. The grouping procedure continues until the least similar pair is reached. We conduct the same grouping process also on the effect side of causal relations.

Note that since the grouping starts from the pair of causes having the highest similarity, it thus forms groups with high inner-similarity within the group members, which guarantees the generated concept (or cluster) to be correct and pure. Another advantage compared to other clustering methods is that there is no need for pre-determining the number of clusters as in k-Means (k clusters) or Hierarchical Clustering (degree of cut). For providing additional clarification, the entire procedure is also described in Algorithm 3 in detail.

Step 2. After grouping cause and effect words respectively, the final score of the causal strength between a cause cluster and an effect cluster is computed as the sum of the global implication scores between the clusters' members normalized by the total number of possible links between both the clusters. Eq. 5.8 shows the way to compute the aggregated implication between a group of causes, C and a group of effects, E.

$$
\begin{aligned}
I_{concept}(C,E) &= \sum_{c\in C, e\in E} I^{[t_s,t_e]}_{global}(c,e) \times \frac{\sum\limits_{c\in C, e\in E} \Gamma(I^{[t_s,t_e]}_{global}(c,e))}{num(C)\times num(E)} \\
\Gamma(I^{[t_s,t_e]}_{global}(c,e) &= \begin{cases} 1 & if I^{[t_s,t_e]}_{global}(c,e) > \sigma \\ 0 & otherwise \end{cases}
\end{aligned}
\tag{5.8}
$$

Here, σ denotes a threshold for creating a link between a cause word and an effect word. $num(C)$ and $num(E)$ are respectively the number of terms in C and the number of terms in E.

Grouping by Similar Patterns. As the second aggregation approach, we propose a method based on the simultaneous consideration of the node similarities on both sides (on the sides of the cause and effect). In contrast to the previous aggregation method, here, we reverse the order of computation so that first we compute the updated score of a given cause-effect pair (step similar to step 1 in the previous method) and then we merge similar pairs (step similar to step 2 in the previous method). The underlying intuition here is that a pair bound by the cause-effect

Algorithm 3 GroupSearch(G)

Input: : Similarity graph of causes G_{cause} (or effects G_{effect}).
Output: Cause groups $C_{group} = \{C_0, C_1, ..., C_m\}$ (or Effect groups $E_{group} = \{E_0, E_1, ..., E_n\}$).

/* Sort the pair of terms by their semantic similarity from highest to lowest. */
Pairs $\{(c_i, c_j), (c_i, c_k), ..., (c_j, c_k)\} \leftarrow SortSimilarity(G_{cause})$
$C_{group}\{C_0 \supset (c_i, c_j), ..., C_u \supset (c_j, c_k)\} \leftarrow IniGIdx(Pairs)$
/* Initialize group index. */
$CurrGIdx \leftarrow 0$
for each $pair(c_i, c_j) \in Pairs$ **do**
  $\{c_k, c_p, ..., c_q\} \leftarrow CommonNeighbors(c_i, c_j)$
  for each $c \in \{c_k, c_p, ..., c_q\}$ **do**
    $GIdx(c_i, c_j, c) \leftarrow Minimum(CurrGIdx, GIdx(c_i), GIdx(c_j))$
  end for
  $CurrGIdx \leftarrow CurrGIdx + 1$
end for

relationship should receive high score if there are many similar cause-effect pairs. For this, we build a graph where each node contains a single cause-effect pair and the link weights represent similarities between different pairs. We show a simplified example of such a graph on the left hand side of Fig. 5.7.

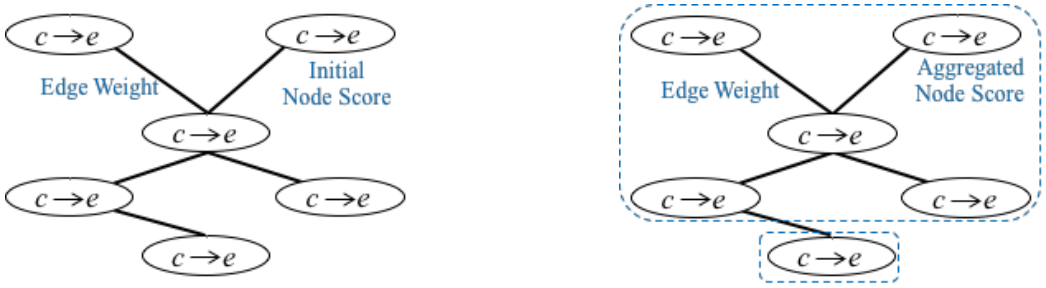


Figure 5.7: Aggregation method by similar patterns.

Formally, let $G = (V, E)$ be an undirected graph composed of the set of vertices V and the set of edges E. Each initial causal relation $(c \rightarrow e)$ is associated with a single node and is assigned with the initial score of its underlying relationship (as computed by Eqs. 5.6 and 5.7). An edge in G is constructed if the two causal relations (v_i, v_j) are similar more than the predefined threshold (0.5 by default). We calculate the node-to-node similarity as the sum of the semantic similarities of both cause and effect sides. Let $Neigh(v_i)$ be the set of vertices that link to a vertex v_i. The score of a vertex v_i, denoted as $Aggr(v_i)$, is computed in a way similar to TextRank algorithm [72] as shown in Eq. 5.9:

$$Aggr(V_i) = (1-d) + d \times \sum_{V_j \in Neigh(V_i)} \frac{w_{ji}}{\sum_{V_k \in Neigh(V_j)} w_{jk}} Aggr(V_j) \tag{5.9}$$

where d is a damping factor set by default to 0.85 and w_{ji} is weight of an edge between two causal relations v_i and v_j. Once each node has its score updated, we group nodes by merging each node with its adjacent nodes retaining the largest clusters. For example, for the two clusters $\{v_1, v_2, v_4\}$ and $\{v_1, v_4\}$, we retain only the larger cluster $\{v_1, v_2, v_4\}$. Within a given cluster the causality strength between the group of causes, C, and the group of effects, E is estimated by the maximum score among all the pairs of causes and effects within C and E, respectively:

$$I_{pattern}(C, E) = \max_{c \in C, e \in E} Aggr(c \rightarrow e) \tag{5.10}$$

The right hand side of Fig. 5.7 contains nodes aggregated into clusters following the above-discussed procedure.

5.4.4 Aggregating Binary Causal Relations by Co-Causal Relations

Besides the need for grouping semantically similar causes and similar effects, another problem arises when two or more different causes collaboratively "co-cause" a certain effect. For example, both `iPod`→`jogging` and `earphones`→`jogging` are found to be causal relationships; however, iPod and earphones are quite different technologies. Therefore, the methods proposed in Sec. 5.4.3 are ineffective to capture this kind of co-causal relation, since the two physical features have quite different meanings, and hence, they should not be treated as belonging to the same concept. The question arises then whether we can aggregate causes to output [`iPod`, `earphones`]→`jogging` and under which conditions such co-causality can be confirmed. Note that we utilize here square brackets to represent the co-causal relationship, in contrast to angular braces used for representing concepts (i.e., the sets of semantically similar terms) as described in Sec. 5.4.3. To detect the co-causal relations, we first define the notion of co-causality.

Definition 5. *Co-causality*. The variables c_i and c_j are said to co-cause the effect e if:

(1) c_i and c_j alone are not sufficient but are necessary to cause the effect e (see the dashed lines in Fig. 5.8) and,

(2) c_i and c_j together raise the probability of the occurrence of the effect e (see the blue arrow in Fig. 5.8).

To implement the notion of co-causality and to group the co-causing terms, for each effect term e, we select a list of candidate cause terms to be used for the co-causality test. As mentioned

above, the cause alone should be necessary but not sufficient to cause the effect (the first requirement of co-causality). It, thus, means the causal strength of the cause on the effect should be at least above zero. Recall that the binary causal strength is represented as a probabilistic notation (see Eqs. 5.6 and 5.7). Based on the selected candidate causes, we then compute the co-causal strength of every possible pair of causes on the effect. Next, we examine if the co-causal strength of the combination of c_i and c_j on the effect e is higher than the causal strength of either of the causes alone (the individual causal strength of c_i and the one of c_j on the effect). If the combination results in the increased strength of causality then we regard the two causes c_i and c_j as co-causing e. Formally, we test,

$$I_{[t_s,t_e]}(c_i \wedge c_j, e) \begin{cases} \geq I^{[t_s,t_e]}_{global}(c_i, e) \\ \geq I^{[t_s,t_e]}_{global}(c_j, e) \end{cases} \tag{5.11}$$

$$\begin{aligned} I_{[t_s,t_e]}(c_i \wedge c_j, e) &= P_{[t_s,t_e]}(e|c_i \wedge c_j) - P_{[t_s,t_e]}(e|\neg(c_i \wedge c_j)) \\ &= \frac{tf(e \in M_{[t_s,t_e]}(c_i \wedge c_j))}{\sum\limits_{i,j \in V} tf(i \in M_{[t_s,t_e]}(c_i \wedge c_j))} - \frac{tf(e \in M_{[t_s,t_e]}(\neg(c_i \wedge c_j)))}{\sum\limits_{i,j \in V} tf(i \in M_{[t_s,t_e]}(\neg(c_i \wedge c_j)))} \end{aligned} \tag{5.12}$$

If Eq. 5.11 holds it means that c_i or c_j itself is not sufficient but necessary to cause effect e and that c_i and c_j together are sufficient to cause e. Under this condition, we can aggregate $c_i \rightarrow e$ and $c_j \rightarrow e$ into a single relationship $[c_i, c_j] \rightarrow e$. The left part of Eq. 5.11 is used as the combined strength of the causal relationship. Eq. 5.12 gives the detailed calculation of the co-causality strength, which is similar to the one in Eq. 5.5.

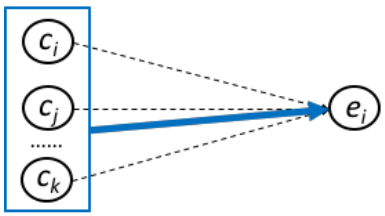


Figure 5.8: Conceptual view of co-causality between multiple causes and a single effect.

Note that there can be also situations when three or more cause terms could actually co-cause the same effect. However, for simplicity, in the experiments, we only focus on detecting cases when two words form co-causal relationships. Note however that our approach can be easily extended to be applied for testing if the higher number of causes is actually co-causing the same effect. This could be done by checking if the co-causal strength of three or more causes is higher than the combinations of two or more causes among these target candidate causes. Algorithm 4 describes the general process of the co-causality detection between multiple causes (two or more) and a single effect.

Algorithm 4 Co-causality Detection

Input: : set of potential causes, C and set of potential effects, E
Output: a ranked list of pair co-causal relations between multiple causes and an effect.

/* 1. Pre-compute global causal strength between each pair of cause and effect. */
$CtoE = \{\}$ /* set of binary causal relations. */
for each $c_i \in C$ **do**
  for each $e_i \in E$ **do**
    add (c_i, e_i) to $CtoE$ if $I_{global}(c_i, e_i) > 0$ (Eqs. 5.6 and 5.7)
  end for
end for
/* 2. Test the causal strength between the combination of causes and an effect. */
$CCtoE = CtoE$ /* set of co-causal relations. */
for $iter = 1...m$ **do**
  for each $(c_i, e_k) \in CCtoE$ **do**
    for each $(c_j, e_k) \in CCtoE$ **do**
      if $I(c_i \wedge c_j | e_k) > I_{global}(c_i, e_k)$ and $I(c_i \wedge c_j | e_k) > I_{global}(c_j, e_k)$ **then**
        remove (c_i, e_k) and (c_j, e_k) from $CCtoE$ add $([ci, cj], e_k)$ to $CCtoE$
      end if
    end for
  end for
end for

5.5 Causality Detection by Simulating "Alternative History"

Lastly, we also investigate additional method for measuring the technology impact. Recall the reasons why token-level causality detection is difficult. Firstly, usually, it is difficult to devise any model for simulating the time series of the effect, as it is typically done in the approaches of Granger Causality and Bayesian Structured methods. This is because, in our problem setting, the effect can be a novel term, such as a new behavior, new situation, new place, which has no historical data to be used for modeling and predicting. Moreover, even if a term is not novel, its use may change over time, while modeling of such changes is not trivial. Secondly, since the temporal values of the effect cannot be modelled, it is then difficult to estimate the counterfactual time series of the effect (i.e., the hypothetical time series without the impact of a potential cause, that is, when the cause would not have occurred).

It is clear that if we could somehow simulate the counterfactual time series of the effect, then we would be able to conduct hypothesis testing for detecting the true cause. Motivated by this reasoning, we propose a new way to simulate the "alternative history" of an effect. The method proposed here will determine the possible effects of a single selected cause. It thus requires selecting the candidate cause as an input, while the ranked list of influenced effects will be delivered as the output.

The approach is as follows. We first remove from the dataset any sentences containing a given selected physical feature or a product, such as iPod. Then we construct the frequency-based time series (or context-based time series) for all the candidate effect terms using such a modified dataset. Note that the time series obtained after the removal of the cause term (e.g., `iPod`) are just the counterfactual time series of each effect without the impact of the cause (e.g., `iPod`). Then to measure the influence of the cause such as iPod on the candidate effect terms, we conduct a hypothesis test (T-test) to estimate if the time series of an effect term (e.g., `jogging`) under the dataset without iPod is significantly different from its corresponding time series derived from the unchanged dataset (one containing all the product models including iPod). Algorithm 5 gives an overview of the process of computing causal strength by "simulating the alternative history". The details of the process behind our approach are also illustrated in Fig. 5.9 and 5.10. Note that in Fig. 5.9, the alternative history of the effect term is simulated by using frequency-based time series to measure the popularity change. Instead, in Fig. 5.10, the context-based time series is used to estimate the change in term usage.

We experiment with this method on the example of `iPod`. Note that in this example we actually use the name of a product model. However, it is possible to use instead some term indicating physical feature. We conduct the simulation experiment on both the verbs and situations considered as conceptual terms (the effect terms). We can discover how the iPod did actually influence our lives quite much. In the results obtained using the frequency-based approach, we can observe verbs such as `downloading`, `running`, `exercising`, `jogging`, `travelling` as well as `gym`, `outdoor`, `travel`, etc. The latter are situations being influenced by the `iPod`. These terms indicate the impact of `iPod` on activities and sports by increasing the occurrence of or enabling some actions, or increasing the frequency of usage or allowing being used in corresponding situations. On the other hand, in the results obtained when using the context-based time series, we can observe such verbs as `charging`, `running`, `driving`, `played` being influenced by `iPod`, which means `iPod` is more often associated with these actions. We can also detect the situations of `iPod` being used, such as `home`, `car`, `gym`, etc.

The above method is an interesting solution as it performs a kind alternative history simulation. That is, it tackles the question on how would the dataset (or our "closed world") look like if a given word had not existed at all. However, the above approach has following drawbacks. One is its poor scalability since it takes considerable time for calculating the counterfactual frequency of effect terms by removing each candidate cause term. In addition, this method requires training the whole corpus to obtain new word embeddings and constructing the context-based time series for each word.

Algorithm 5 Overview of Causality Detection by Simulating Alternative History

Input: : selected time series dataset (e.g., 20 years review of portable music device); a cause term (e.g., iPod); a set of potential effects

Output: a ranked list of effect terms impacted by the given cause term.

1: Compute the frequency- and context-based time series for each potential effect using the input time series dataset.
2: Create the modified dataset by removing the content related to the cause term (e.g., removing the sentences which contain the cause term);
3: Re-compute the frequency- and context-based time series for each potential effect using the modified dataset.
4: For each effect term conduct hypothesis test (e.g., T-Test) to determine if its modified time series (either frequency- or context-based) differs significantly from the original corresponding time series.
5: Add the effect term to the returned list if the removal of the cause term significantly influences its time series.

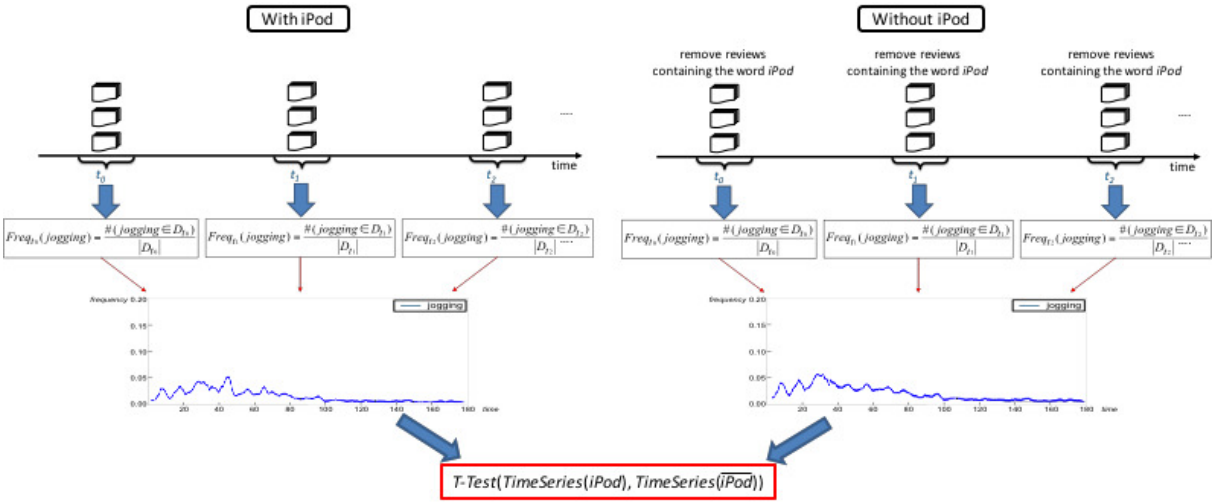


Figure 5.9: Measuring the influence of term iPod by "simulating the alternative history" based on popularity change.

5.6 Experiments

5.6.1 Dataset

We conduct experiments on the Amazon Product Review Dataset [71] which is provided by the Stanford Network Analysis Platform (SNAP). It covers 18 years since June 1995 up to March 2013 and includes more than 34 million reviews about over 2.5 million products. The products are organized into the hierarchy of categories. For the experiments, we choose three large sub-categories of Electronics: Portable Audio & Video, Electronics, Computers & Accessories, Laptops and Electronics, Camera & Photo. We have chosen these categories as many important changes occurred within the past decade in relation to the products described by these categories.

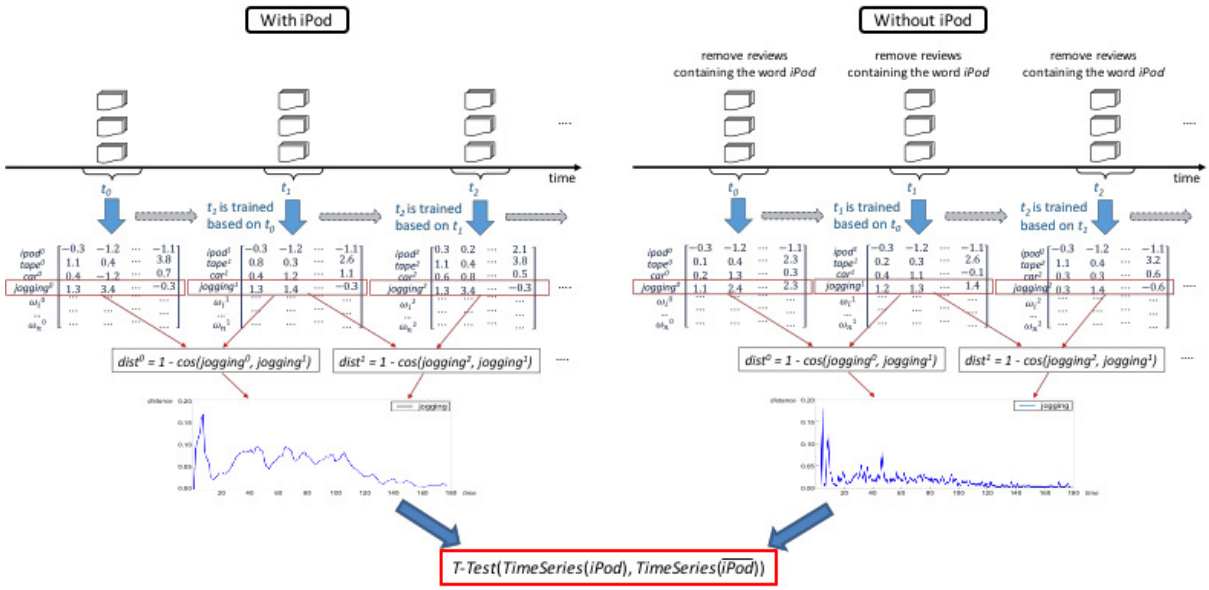


Figure 5.10: Measuring the influence of the term iPod by "simulating the alternative history" based on contextual change.

Table 5.1: Statistics of Evaluated Categories.

Category	Num. of Models	Num. of Reviews	Time Span
Electronics, Portable Audio & Video	7,809	182,831	2000-2012
Electronics, Camera & Photo	21,289	289,956	1999-2012
Electronics, Computers & Accessories, Laptops	1,021	7,814	2000-2012

The statistics of the selected categories are summarized in Table 5.1.

5.6.2 Feature Extraction

First, we need to extract candidate cause and effect terms as briefly mentioned in Sec. 5.2. In the current implementation, we use verbs (e.g., `navigate`, `scroll`) and situation terms (e.g., `gym`, `home`) as effect terms. We automatically detect verbs using POS tagger. As for the situation terms, we extract the nouns appearing directly after prepositions (e.g., `in`, `during`). Note that it is also possible to utilize existing vocabulary lists for selecting situation terms such as the lists of locations.

On the other hand, automatically retrieving physical product features requires more processing. The problems lie mainly in that: (1) physical product features vary across different products, therefore, we cannot use any predefined or general feature lists while still not missing any unique characteristics of products. Another issue is that (2) some physical terms are novel (e.g., `USB`, `mp3`) which may not exist in commonly used vocabulary lists or dictionaries. It is thus difficult to define the meaning of such new words by using existing dictionaries.

Considering these challenges, we propose classification model to distinguish if a term is a

physical product feature or not. As classifier features, we make use of both the semantic meaning of words and their characteristics derived from a lexical database such as WordNet. The types of classification features we use are as follows:

1. *Word Semantics*: We use the distributed representation [74] of a word to represent its meaning because similar terms should belong to the same class. To capture term meaning, we train fixed word embeddings by using all the reviews of the target category regardless of their time stamps. After training, similar terms, such as `cassette`, `CD` and `mp3`, will be positioned close to each other in the vector space (since they all represent the same concept of storage medium). This means they are more likely to belong to the same class. We set the number of dimensions for word embeddings to 100.

2. *Lexical characteristics*: We select 5 classifier features derived from the WordNet as follows.

   a *Distance to the node "physical entity" (integer)*: We measure the distance to the node physical entity in the WordNet hierarchy. The smaller the depth is, the more physical the word is likely to be.

   b *Distance from the node "abstraction" (integer)*: We measure the distance from the node abstraction to the target word within the WordNet hierarchy. The higher the distance, the more physical (less abstract) the word is likely to be. This is because physical product features tend to be represented by concrete, rather than, abstract words.

   c *Number of hyponyms (integer)*: General words tend to have on average more hyponyms. Since physical product features are usually more specific, then we assume that the fewer hyponyms a word has, the more physical it is likely to be.

   d *Similarity to physical product feature markers (float)*: We adopt here the WordNet Similarity measure [85] to calculate the semantic similarity between a given word and a set of fixed markers of physical product features: "size", "weight", "color", "shape", and material. The assumption is that the higher the similarity is, the more physical the word is likely to be.

   e *Plural form (binary)*: In many cases, if a word can be expressed in plural form, then it is more likely to be a physical product feature such as batteries, cases, etc.

We train SVM classifier with linear kernel and default settings using 250 manually tagged terms and then we evaluate its performance through 5-fold cross validation. Table 5.2 shows the precision, recall, F1-score and accuracy of the classifier when applying all the types of features.

Table 5.2: Evaluation of SVM Classification Model.

Feature selected	Precision	Recall	F_1-score	Accuracy
Semantic + Lexical	**0.874**	0.863	**0.865**	**0.859**
Semantic	0.833	0.812	0.81	0.804
Lexical	0.67	**0.957**	0.788	0.726

The results indicate that considering both the semantic meaning and lexical features results in the highest performance in terms of precision (0.874), F_1-score (0.865) and accuracy (0.859). We thus use both semantic and lexical features for extracting physical product features.

5.6.3 Analyzed Methods

We describe here the baselines and the proposed methods to be tested.

Baselines. We prepare three baselines as follows:

(1) **Jaccard Coefficient** (**Jacc_Coef**): in this method we first detect change periods of the time series of a given candidate effect term. Then within each such change period, we directly compare Jaccard Coefficient score between any potential cause term and a target effect term. The cause term which has the highest Jaccard Coefficient score will have the highest causality strength. We apply this baseline to examine whether the co-occurrence could be sufficient to estimate the causality between terms.

(2) **Lasso Granger Causality** (**Lasso_GC**) is the stronger baseline that we adopt. It has been proposed by Arnold et al. [5] to compute causality strength based on the theory of Granger Causality. Lasso Granger Causality is its modification that alleviates the computational problem of Granger Causality by applying Lasso algorithm, which embodies a method of variable election using L_1-penalty term. The idea is to identify the subset of causes, given the fact that the best regression for that variable with the least squared error will, in theory, have non-zero coefficients only for the variables in the neighborhood. Lasso Granger Causality is widely adopted to detect the type-level cause-effect relationships. For example, it has been used to detect causal relations between variables within numerical dataset (e.g., the causal relation between real gross domestic product (GDP) and real gross domestic savings (GDS) for Morocco). We use it in our experiments to answer the question whether the type-level causality detection methods could be suitable for our task (i.e., tackling the token-level causality detection).

(3) **Graphical model - Dynamic Bayesian Networks** (**GraphicalDBN**) is one of the Bayesian approach causality models. In the experiments, we use the state-of-the-art implementation proposed by Brodersen et. al [14], called *CausalImpact*[1], which detects the degree of the impact of a change or an intervention (e.g., advertisement) on an existing time series (e.g.,

[1] `http://google.github.io/CausalImpact/`

sales of a product). The key idea behind this approach is to construct a model for predicting the counterfactual time series which measures how the response metric would have evolved after the intervention time point if the intervention had not occurred. Next, if the predicted time series is significantly different than the real one, the intervention can be then determined as a cause. For example, *CausalImpact* could be used to gauge whether adding a new feature of a product caused an increase in the number of app downloads.

Proposed Methods. We test four proposed methods as below.

(1) **Initial Causality (IniCausal)**: this method (see Eq. 5.5 in Sec. 5.4.2) is regarded as the basis for the concept of detecting causality between two words over time.

(2) **Global Causality (GlobCausal)**: this method (see Eq. 5.6 and 5.7 in Sec. 5.4.2) is applied to test if considering the global information of all the candidate causes of a given effect can remove spurious causes.

(3) **Aggregation based on Similar Concept (AggrConcept)**: this method (see Sec. 5.4.3) is used for testing if the aggregation by similar concepts helps to generate better results. This aggregation groups both the causes and effects separately and outputs the cause-effect results in a cluster format.

(4) **Aggregation based on Similar Pattern (AggrPatt)**: we apply this method (see Sec. 5.4.3) to test if the aggregation by similar patterns performs differently from the aggregation by similar concepts. In other words, we examine if it is better to aggregate both sides of relations at the same time.

(5) **Aggregation based on Co-Causality (Co_Causality)**: we apply the method described in Sec. 5.4.4 to test the effectiveness of detecting different causes that lead to the same effect. Note that the Co_Causality method is orthogonal to the methods **AggrConcept** and **AggrPatt**. In this sense, we cannot compare Co_Causality with other methods by means of the evaluation. We will show then the quantitative scores of Co_Causality separately in Sec. 5.7.2.

5.7 Evaluation

We conduct both quantitative and qualitative evaluation. Their results are described in this section.

5.7.1 Quantitative Evaluation

Test Sets As far as we know, no ground truth data is available for the task of token-level causality detection within temporal document collections (i.e., detection of causal relation such that the change in one word implies the change of another word). We have thus manually created test sets containing cause and effect pairs that existed within the time span of each category utilizing external resources including Wikipedia, several dedicated websites [98, 89, 99, 95, 92,

100, 94, 97, 91, 90, 96, 93] and a Web search engine. We prepared 54 cause-effect pairs of the category Electronics, Portable Audio & Video and 56 pairs for the category Electronics, Camera & Photo considering their corresponding time spans. The ground truth data contain two types of effects: actions and usages. Actions are described by verb phrases while use situations are described by nouns such as location terms. Table 5.6 shows examples of ground truth patterns for the category Electronics, Portable Audio & Video. We will consider the output causal pair as correct if both its cause and effect sides are semantically similar to the corresponding sides in any of the ground truth cause-effect pairs. Note that in the ground truth, the effects are sometimes described by verb phrases (e.g., "watch movies"), while the tested methods output either verbs or situation terms as effects. Therefore, when the detected effect is in the form of a verb (e.g., `watch`), we consider it to be correct if its underlying verb matches any verb in the ground truth.

Evaluation Measures For evaluation, we output up to 10 top results for each year. Then, we combine the results generated for all the years and compare with the ground truth. We compute precision, recall and F_1-score to measure the performance of each method.

Since we make use of two types of time series (frequency-based and semantic-based), we generate the results for each type separately and we evaluate them separately (see columns "Frequency-based" and "Semantic-based" in Tables 5.3-5.4). In addition, we also evaluate the performance when combing the results coming from the two types of time series (see column "Freq.-based + Sem.-based" in Tables 5.3-5.4). In order to keep the number of returned results the same for different approaches, we merge in each year the top 5 results returned by the method using Frequency-based and the one using Semantic-based time series.

Evaluation Results Tables 5.3-5.4 describe the performance of each analyzed method. We notice that the proposed methods **AggrConcept** and **AggrPatt** outperform the two baselines over all the metrics, which proves the proposed approach performs well. We list the other findings below:

(1) Co-occurrence is not enough for measuring causality. According to Tables 5.3-5.4, the causality detection is quite difficult as evidenced by quite poor performance of **JaccCoef**. This suggests that although the co-occurrence describes the relatedness of two terms, it fails to capture the causation (the cause must be a necessary condition for the effect [58]).

(2) Time series analysis is not enough for measuring causality within text. **LassoGC** is the typical method for computing the causality between time series. It however delivers poor results when applied for discovering the causality within text. Note that unlike **LassoGC** our proposed methods take into consideration both the time series and the probability of term occurrence within text. In contrast to continuous data (e.g., humidity, GDP), where **LassoGC** is typically applied,

Table 5.3: Results for the Category Electronics, Portable Audio & Video.

Method	Frequency-based			Semantic-based			Freq.-based + Sem.-based		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
JaccCoef	0.11	0.18	0.14	0.11	0.18	0.14	0.11	0.18	0.14
LassoGC	0.14	0.22	0.17	0.14	0.22	0.17	0.18	0.28	0.22
IniCausal	0.18	0.28	0.22	0.11	0.18	0.14	0.21	0.34	0.26
GlobCausal	0.2	0.32	0.25	0.19	0.3	0.23	0.29	0.46	0.35
AggrConcept	0.24	0.38	0.3	**0.28**	**0.44**	**0.34**	**0.36**	**0.58**	**0.45**
AggrPatt	**0.28**	**0.44**	**0.34**	0.21	0.34	0.26	0.34	0.54	0.42

text tends to be more arbitrary and complex. Relying solely on the time series analysis is thus not sufficient for detecting causal patterns in text collections.

(3) Computing causality over all candidate causes is necessary. **GlobCausal** has been found to be consistently more effective than **IniCausal**. This signals that some spurious or weak causal relations are removed by additional filtering that retains genuine causes (see Eqs. 5.6-5.7).

(4) Aggregation process helps to validate and group cause-effect patterns. As we discussed in Sec. 5.4.3, by aggregating semantically similar binary causal relations, we provide more evidence of the actual causality. For example, the pair `iPod`→`jogging` is returned at the 68$^{\text{th}}$ rank by the method **GlobCausal**, while **AggrConcept** and AggrPatt return it within the top 5 results. In addition, the grouped similar cause-effect relations have better explanatory power. Another observation is that `AggrPatt` performs better than **AggrConcept** when using the frequency-based time series.

(5) Frequency-based and semantic-based time series complement each other. Although methods using both the frequency-based and semantic-based time series are generally effective, their combination helps to discover more correct cause-effect relations than when used alone. This is because there is relatively small overlap between their outputs. Thus, we can say the approaches based on these time series complement each other. This is demonstrated by the improved performance when combing the results from the methods based on each of the time series. The frequency-based time series approaches allow discovering frequent relations. On the other hand, the semantic-based ones help to find the causality between components not commonly mentioned in the dataset, yet, subject to semantic change (i.e., change of the context in which a word is used). This can be observed by analyzing example results in Table 5.6 (IDs: 18, 23, 26, 33 and 46) which are found by applying methods that utilize the semantic-based time series. The verbs store, share, delete, surf and navigate are verbs indicating new kinds of actions that became available following the advent of `mp3`, `Napster` and `iPod`.

Table 5.4: Results for the Category Electronics, Camera & Photo.

Method	Frequency-based			Semantic-based			Freq.-based + Sem.-based		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
JaccCoef	0.1	0.16	0.12	0.1	0.16	0.12	0.11	0.18	0.14
LassoGC	0.16	0.26	0.2	0.23	0.36	0.28	0.26	0.34	0.29
IniCausal	0.13	0.2	0.15	0.21	0.34	0.26	0.3	0.48	0.37
GlobCausal	0.15	0.24	0.18	0.29	0.46	0.35	0.35	0.56	0.43
AggrConcept	0.16	0.26	0.2	0.38	0.6	0.46	0.44	0.7	0.54
AggrPatt	**0.21**	**0.34**	**0.26**	**0.43**	**0.68**	**0.52**	**0.46**	**0.74**	**0.57**

Table 5.5: Types of Environment Settings.

Settings	Description
freq_verb	Using frequency-based time series to detect causality between physical features and verbs
freq_sit.	Using frequency-based time series to detect causality between physical features and situation terms
sem_verb	Using semantic-based time series to detect causality between physical features and verbs
sem_sit.	Using semantic-based time series to detect causality between physical feature and situation terms

5.7.2 Qualitative Evaluation

To further evaluate the quality of the results we also conducted user-based analysis. We invited 5 subjects (2 males and 3 females in their 30s) to annotate the results using several quality criteria.

Settings Before describing the results, we first clarify the evaluation settings. We utilize the detected physical product features as potential causes, while the extracted verbs and situations are regarded as two types of effects. We also generate two types of time series as described in Sec. 5.3: the frequency-based and the semantic-based time series. So, in total, we have 4 environment settings considering possible combinations of the effect types and time series types. These are summarized in Table 5.5.

As discussed above, we have 7 methods to be tested (4 proposed methods and 3 baselines). The cause-effect results by every combination of the method and environment settings are then returned for each of 11 years (the time period when the reviews in the category Electronics, Portable Audio & Video of the Amazon Product Review Dataset were created).

The annotators were asked to evaluate the results generated for each year by the 28 approaches (7 methods, each with 4 environment settings). Note that the top 10 results with the highest causality strength are returned on average for each year by every analyzed method. The criteria of the evaluation consist of 3 dimensions: correctness, novelty and comprehensibility (as described in the next section). Each annotator thus gives: 11(years) * 4(environments) * 7(methods) *

4(dimensions) = 1,232 scorings.

Evaluation Criteria We have come up with 4 criteria based on the general notion of usefulness of the cause-effect relationships that ideally should be correct, diverse, interesting and understandable for users to effectively make sense of the results. These criteria are described as follows.

- **Correctness**: by this measure we wish to analyze how sound the results generated for each year are. During the assessment users are allowed to utilize any external resources they feel are useful, such as Wikipedia, Web search engines, books, or particular online sources, etc. We consider the correctness measure as the most important.
- **Novelty**: it measures how novel the results are within the same year. In other words, it quantifies how varying and diverse information the annotators could acquire after viewing the results at a given year.
- **Unexpectedness**: we assumed that many results can be obvious, so we decided also to measure how easy would it be to come up with the results by a user herself or himself. We assume that too obvious results will not be useful and informative for users. This score is similar to the notion of rule interestingness in the association rule mining.
- **Comprehensibility**: it measures how easy it is to understand and explain the results. On the one hand, grouping words into clusters should provide more information on the actual meaning of the concept underlying the cluster. On the other hand, it depends on the effectiveness of clustering as the introduction of noise may impair users understanding of the results. We then need to test how easy to understand the output rules are, and whether (or how much) the aggregation actually contributes to improving the comprehensibility of causal relationships.

All of the scores were given in the range from 1 to 5 (1: not at all, 2: rather not, 3: so so, 4: rather yes, 5: definitely yes).

Evaluation Results **Correctness.** We first focus on the key criteria of the evaluation, correctness. In Fig. 5.11 we demonstrate the average correctness scores for each combination of methods and the used experimental environments. The following observations can be made. First, we notice that the proposed method **AggrPatt** achieves the best performance among the results by all the methods over every environment. It gives the optimal results when applied on context-based time series with verbs being used as candidate effects (sem_verb). In this setting **AggrPatt** outperforms the strong baseline **LassoGC** by 10.7%. Note that the weaker baseline,

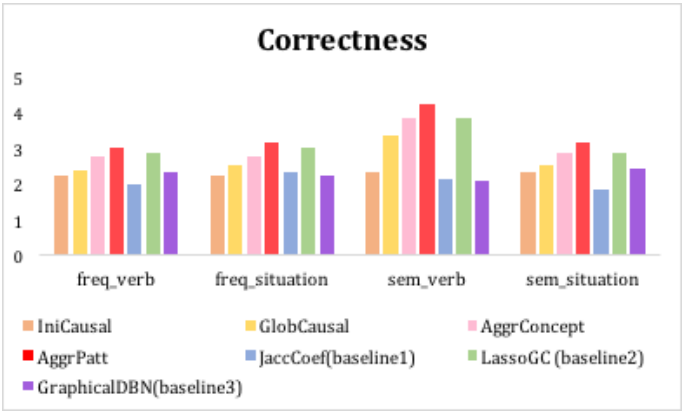


Figure 5.11: Evaluation of Results based on Correctness.

JaccCoef, somewhat, not surprisingly, has the worst performance over all the environments. This clearly demonstrates that naive co-occurrence based approach is not sufficient to be applied as a causality measure. As for the other baseline, **GraphicalDBN**, we can observe that it achieves even lower performance than **LassoGC**. This might be due to difficulty in expressing causes by the time series of token-level effect using Bayesian model, which simulates the local trend, seasonality and other patterns of the time series. However, as mentioned before, the occurrence of a new behavior or a social activity tends to be somehow exceptional case, which usually has not occurred before or its previous occurrences were under different circumstances. In addition, users may mention some novel action/situation in online reviews during the time when the new feature occurs, while they may mention them less after the feature has been widely adopted. As a result, **GraphicalDBN** achieves even worse performance than **LassoGC** in measuring the influence of a cause on the effect. We also notice that **AggrPatt** performs better than **AggrConcept** by 10.8% on average, which means that the concept of simultaneously aggregating causes and effects by grouping similar cause-effect patterns is useful and the aggregation is more efficient than when outputting separate results. Actually, as we will see later, the former works the best within all the 4 evaluation criteria, which proves the utility of aggregation. **AggrConcept** achieves better results than **LassoGC** when applied on the context-based time series; yet, we notice it is weaker in the frequency-based settings. The weaker performance in the frequency domain may be explained by the observation that **LassoGC** performs well in detecting frequent causal patterns which are common and correct causal relations such as `CD→burn` and `MP3→download`. **GlobCausal** is consistently more effective than **IniCausal** (by 17.2% on average), which signals that some spurious or weak causal relations could be removed by applying additional filtering such as the one that contrasts all the candidate causes with each other. Actually, as we will see later **GlobCausal** is superior than **IniCausal** based on the remaining criteria, too.

Novelty. Fig. 5.12 describes the average novelty scores for all the methods. This time we ob-

serve that both the proposed methods based on aggregation procedure, **AggrPatt** and **AggrPatt**, manage to outperform our baselines in each different setting. Surprisingly, even **GlobCausal** is better this time than the baselines when applied on the context-based time series. On average it outperforms **JaccCoef** baseline by 57.3%, **LassoGC** baseline by 6% and **GraphicalDBN** baseline by 27.3%. The best method according to the diversity criterion is again **AggrPatt** that outperforms the best performing baseline **LassoGC** by 18% in the experiments using the frequency-based time series, and by 41% in the experiments when utilizing the context-based time series. The poor performance of **LassoGC** under the diversity criterion clearly shows the limitation of popular Granger Causality approach in detecting the token-level causality. It tends to prefer dominant causes (e.g. `CD`, `MP3`, `iPod`, etc.) to explain the effects, yet, it ignores infrequent, but, important causes. In other words, this baseline method is characterized by weak recall.

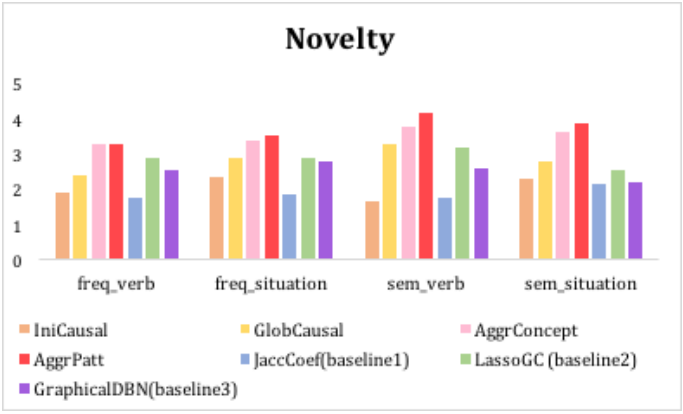


Figure 5.12: Evaluation of Results based on Novelty.

Unexpectedness. Next we turn our attention to the uniqueness of the results. In Fig. 5.13, we evaluate the generated results according to the unexpectedness criteria. It can be noticed that almost all our proposed methods outperform the baselines including the strong baseline, **LassoGC** which, under this criterion, consistently obtains quite poor evaluation across all the environments used in the experiments. For example, **AggrPatt** has almost two times better scoring than **LassoGC**. As mentioned before, **LassoGC** tends to output common cause-effect patterns. Its results tend thus to be often obvious. On the other hand, our proposed methods are capable of not only finding explicit cause-effect relationships but they also manage to detect more implicit relationships, which can provide more interesting and novel information to users. Similar to **LassoGC**, **GraphicalDBN** tends to detect many obvious and trivial relations such as `battery`→`charged`, `battery`→`replaced` and `battery`→`powered`.

Comprehensibility. Lastly, in Fig. 5.14 we compare the comprehensibility scores of the returned results. What can be observed from this table is that both the aggregation methods **AggrConcept** and **AggrPatt** achieve higher performance than the other methods. We thus conclude

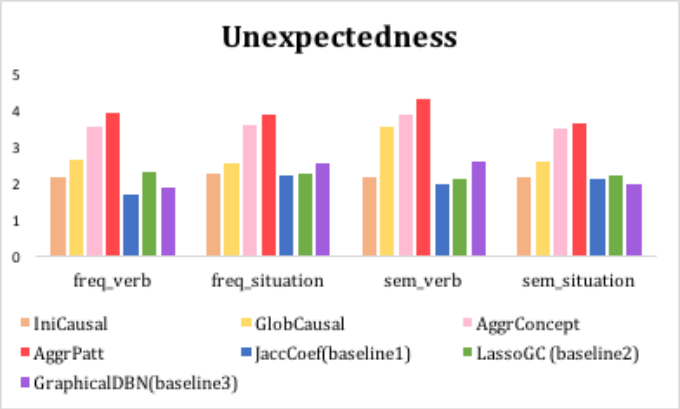


Figure 5.13: Evaluation of Results based on Unexpectedness.

that they can help users to better understand and to make more sense from the generated causal relationships. Especially in the semantic task using the context-based time series (sem_verb and sem_situation) the difference appears to be quite high (55% on average). Interestingly, **GlobCausal** manages to outperform in such settings the stronger baseline, **LassoGC**, by 22% and the weaker baseline, **GraphicalDBN**, by 35%.

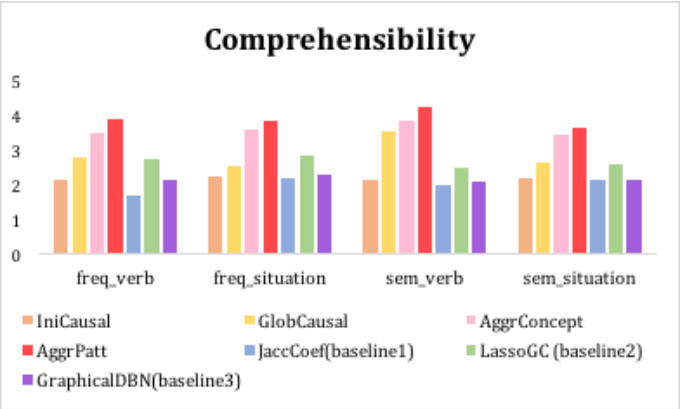


Figure 5.14: Evaluation of Results based on Comprehensibility.

Evaluation Results of Co-causality Approach. In this section, we evaluate the results of Co_Causality aggregation that was described in Sec. 5.4.4. We test in this case all the three categories used in our experiments: Electronics, Portable Audio & Video, Electronics, Camera & Photo and Electronics, Computers & Accessories, Laptops. For each category, we conduct 2 experiments of co-causality by treating verbs or situation words as effect terms, respectively, under the frequency-based time series. The results are evaluated based on the 4 above-discussed criteria and are shown in Fig. 5.8. Blue color denotes the results for the case when verbs are used as conceptual terms while yellow color is when situation words are utilized. We see that the results differ among different categories and are on average the lowest for the Electronics,

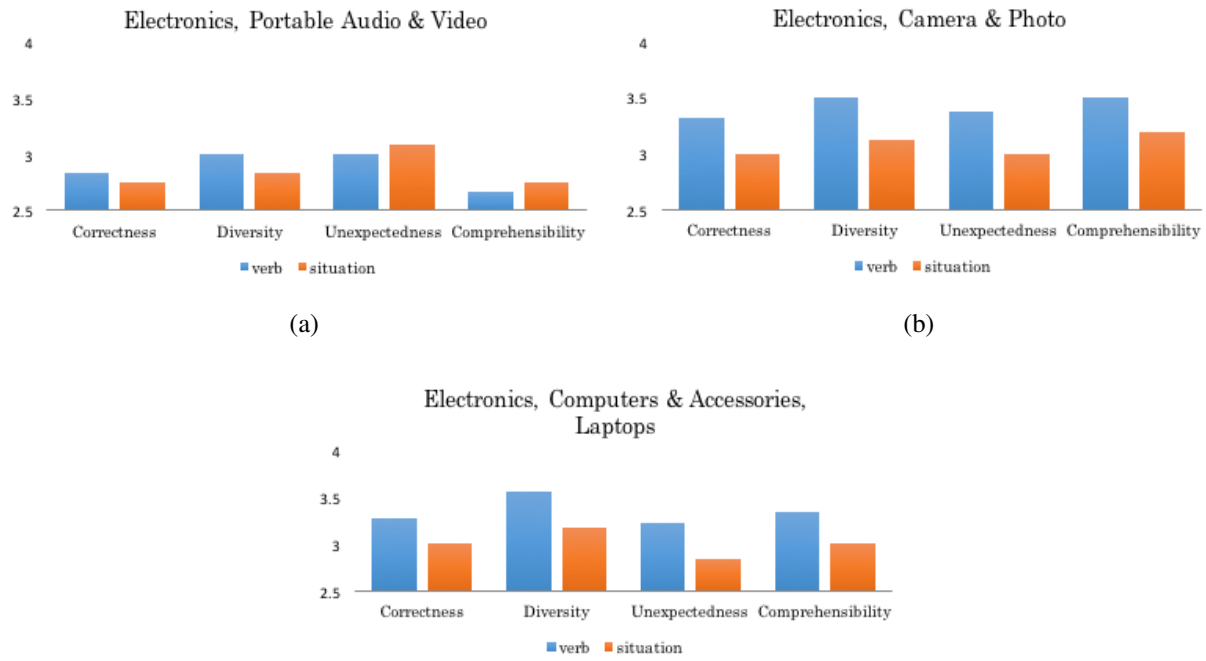


Figure 5.15: Evaluation of results of the co-causality aggregation based on the Correctness, Diversity, Unexpectedness and Comprehensibility. (a) shows the results for the category "Electronics, Portable Audio & Video". (b) demonstrates the results for the category "Electronics, Camera & Photo", and (c) illustrates the results obtained for the category "Electronics, Computers & Accessories, Laptops". Blue color denotes the results when verbs are used as conceptual terms, while red color is when situation words are utilized.

Portable Audio & Video category. Usually, using verbs achieves better results than when using situation words as reflected in consistently high score of the former in different categories.

5.7.3 Case Studies

In this section, we analyze in detail selected results obtained for the category "Electronics, Portable Audio & Video". We choose this category as it has many changes that occurred within the time scope of the analysis and which are useful for discussion. We will also pick up several examples of causal relations that were found in the other two product categories.

As discussed in Sec. 5.4.3, the aggregation procedure helps to cluster similar cause-effect relationships in order to generalize word-based cause-effect relations into relations binding concepts. When seeing the results of **IniCausal** or **GlobCausal**, users only can observe the relations between single words such as `mp3→disply` and `mp3→jogging`. On the other hand, the aggregation methods, **AggrConcept** and **AggrPatt** allow understanding more general and meaningful implications like ones suggesting that (a) the innovation of the storage media `CD` and `MP3`

enabled to listen to music when performing sports, (b) it allowed displaying lyrics in panel, and (c) it made it possible to play and to record music.

The next observation is that both the context-based and frequency-based time series are useful for discovering the cause-effect relations which are often complementing each other. The frequency-based time series obviously results in discovering frequent patterns, while the context-based ones can find the causality pairs that may not often be evident and directly supported by the data, yet, that were driven by the contextual changes (i.e., change of the context in which a word is used). We notice more interesting information. For example, when looking at the results by **AggrConcept** and **AggrPatt**, we detect some novel verbs in relation to music devices popular at that time such as `reboot`, `corrupt`, `reinstall`, `syncing`, `tag`, `delete`, `drag`, `shuffle`, `select`, `navigate`, `browsing` etc., which have not appeared before (i.e., within the time period when a `cassette tape` was a dominating storage medium). Those verbs appear due to the subsequent advent of {`CD`, `MP3`} suggesting that new actions and behaviors were possible (i.e., we can obtain the evidence of technology influence on our lives).

Another interesting finding is that by using our approach, we can observe some evolving relationships from the results. We can track how the technology usage changed within the same situations. For example, we have found the technology changed according to the sequence of `radio>CD>MP3>VCR>TV>iPod` in the situation such as `car`. Similar findings can be observed in the category of "Electronics, Camera & Photo". By looking at the cause side, we can see that the storage media of the camera evolves in the chain of {`film`, `tape`}`>sd>card>dvd`. In the category of "Electronics, Camera & Photo", at the very beginning, people utilized {`Kodak`, `film`} to take a photo and they needed to print the photos, while in the latest year, the used cameras became more advanced having lens and binoculars and using digital formats of data.

By detecting the co-causality, we can find some co-occurring necessary technology components to fulfill the implementation of the effect. For example, we detect the co-causality such that [`mp3`, `card`] causes recording. This pair makes sense since `mp3` is only the storage format, while `card` is the physical storage media. `Card` together with `mp3` enable recording to implement. For the category of "Electronics, Camera & Photo", we detect that [`camcorder`, `film`] causes recording. Also we find that [`tripod`, `binoculars`] contributes to stabilization. Finally, [`canon`, `s400`] enables the usage when being underwater etc.

5.8 Additional Discussion

Underlying collection. Generally, the proposed approach is flexible in choosing any level of categories of products. However, enough data is needed to construct both the frequency- and context-based time series. Especially, the latter one requires large amount of data for training

word vector representations at each time point. Another issue which might impact the final results is when exploring the causality over a long time dataset. In constructing the context-based time series, we assume that most of the commonly used terms maintain stable semantics over time. However, in collections spanning long time periods (e.g., several decades) some terms may change their meanings. In this sense, one need to rule out the factor of term meaning evolution.

Choice of candidate terms. Currently, we utilize unigrams for both cause and effect terms. This approach can be naturally extended to bigrams or phrases in order to more elaborately describe social behaviors or situations related to the products use (e.g., jogging in the gym, delete the songs). In the future, other approaches for effective conceptual term selection can be considered such as verb+nouns, paired activities and situations (e.g., a given activity done in a particular place) and so on. This would necessarily involve making changes to our methodology in order to effectively represent the frequency-based and context-based time series. For example, a particular example phrase or a pattern such as listening music in gym will likely have very low mention count over time making it difficult to reason about its popularity and context across time. Therefore, in such a case it is necessary to devise effective techniques for accurately representing temporal fluctuations of longer and more specific expressions (e.g., approaches based on aggregating component time series).

We also apply SVM model to detect product features, specifications or components, which represent new technologies and are likely to trigger some impact on social life. We detect effect terms from verbs and situation terms. This step can be improved using other related techniques.

Related issue is with tracking the changes in attribute values such as the change in price, size, weight, battery life, etc. In certain cases, the change in attribute value may allow for novel usage of products. For example, the size of music devices has been getting progressively smaller to allow storing them in pockets and thus carrying outside. We would like to explore this problem in the future.

Visualization of results. In the current implementation, we output the results as tables containing detected causal relations between words or between groups of words. However, an effective visualization is necessary for providing users with easy-to-understand results to represent the causal relations along the time and, possibly, to offer good level of interactivity necessary for visual analytic.

Predicting future impact. Predicting the occurrence of a novel social behavior caused by a new product or prediction of the degree of impact of the new product on the social life is an interesting yet challenging task. Naturally, the proposed approach allows detecting causal relations within the past time series. How to utilize the detected patterns to predict the potential future behaviors when a new feature appears remains however an open question. The causal

patterns delivered by the proposed methods could serve as a training data for potential prediction approaches.

5.9 Summary

With the increasing number of product reviews left by users over the recent years, it has become possible to automatically extract novel types of knowledge related to technology progress over time. This chapter proposes a novel usage of temporal product review collections. In particular, we propose estimating the effects of technology and product evolution on our lives. We tackle this challenge by introducing the idea of detecting cause-effect relationships involving time series representations of terms. Such detection uncovers implicit causal relations in contrast to the explicit causality in natural language (e.g., causal patterns based on the occurrence of cue phrases such as due to, because of, etc.). To increase the accuracy and to decrease the computational cost we constrain candidate terms to those related to technologies (called physical terms) and those related to product usage (called conceptual terms).

Unlike typical approaches that utilize term frequency over time we also propose to detect contextual changes in words (changes of word context) by using neural network based word embedding. This is achieved by sequentially re-training word representation over consecutive slices of the underlying document collection. Both the temporal term representations provide complementary results as demonstrated in the experiments.

Next, we discuss a novel proposal of relation aggregation to extend the binary causal relations for obtaining more exhaustive and comprehensive causal patterns. One of the aggregation methods we use aims to detect co-causality patterns which occur when two different concepts cause a given effect at the same time. Finally, we demonstrate an additional approach in which we test the changes brought about by removing information on a particular feature or model from the dataset. In particular, we compare neural network based word representations between the complete (i.e., original) and the altered dataset.

The experimental evaluation demonstrates that our methods outperform baselines across all the considered criteria: correctness, diversity, unexpectedness and comprehensibility.

Table 5.6: Example results where Cause and Effect are the ground truth relations. The tags (0, 1) shown in parentheses denote the results using the frequency-based and semantic-based time series, respectively (1 means the results match the ground truth causal relations, while 0 means otherwise).

ID	**Cause→Effect**	**JaccCoef (baseline) (Freq., Sem.)**	**LassoGC (baseline) (Freq., Sem.)**	**IniCausal (proposed) (Freq., Sem.)**	**GlobCausal (proposed) (Freq., Sem.)**	**AggrConcept (proposed) (Freq., Sem.)**	**AggrPatt (proposed) (Freq., Sem.)**
4	radio→car	(0, 0)	(1, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
9	CD player→skip (rewind the track on CD)	(1, 1)	(1, 0)	(1, 1)	(1, 0)	(0, 1)	(0, 0)
10	CD player→recording sound	(1, 1)	(1, 1)	(1, 0)	(1, 0)	(1, 0)	(1, 0)
14	CD player→car	(0, 0)	(0, 1)	(0, 1)	(0, 1)	(1, 1)	(1, 1)
16	CD player→display on panel	(0, 0)	(0, 0)	(0, 0)	(1, 0)	(0, 0)	(1, 0)
18	mp3→store more music	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
23	Napster→share song files	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 0)
26	mp3 player (iPod)→delete songs	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
30	mp3 player (iPod)→watch movies	(0, 0)	(1, 0)	(1, 0)	(1, 0)	(1, 0)	(1, 0)
32	mp3 player (iPod)→jogging	(0, 0)	(1, 0)	(0, 1)	(0, 1)	(1, 1)	(1, 1)
33	iPod→surf the web	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)	(0, 0)
36	mp3 player (iPod)→gym	(0, 0)	(1, 0)	(1, 1)	(1, 1)	(1, 0)	(1, 1)
38	iTunes→download songs	(1, 1)	(1, 1)	(1, 1)	(1, 0)	(1, 1)	(1, 1)
42	iPod→car	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 0)
46	iPod→navigate song lists	(0, 0)	(0, 1)	(0, 0)	(0, 1)	(0, 1)	(0, 1)

CHAPTER 6

CONCLUSIONS

6.1 Summary

This thesis discussed object search and relationship search techniques for more effective exploring of text archives. We addressed the research problem of terminology gap when searching object in an unknown domain as well as the challenges of searching relationships in text archives. Three research topics described in this thesis are summarized as follows:

- **Detecting Semantically Similar Terms across Different Domains**
  This work approaches the problem of finding temporal counterparts as a way to build a "bridge" across different time periods. Knowing corresponding terms across time can have direct application in supporting search within longitudinal document collections or be helpful for automatically constructing evolution timelines. We first discussed the key challenge of the temporal counterpart detection the fact that the contexts of terms significantly change over time. We then proposed the global correspondence method using transformation between two vector spaces (past and present). We demonstrated two effective ways for automatically finding training sets of anchor pairs for building a transformation matrix. Based on global correspondence, we next introduced a more refined approach of computing the local correspondence. Finally, we proposed a method for correcting OCR driven errors as a post-processing step and introduced new approach for explaining and visualizing results. We conducted the experiments on two datasets: one is the New York Time annotated corpus which contains 20 years long set of news articles; the other is much longer news articles dataset, the Times Archive, which contains 200 years' long collections of digitized news papers. Through experiments we demonstrated that the local correspondence using global transformation with semantic stability constraint outperforms both the baselines and

the global correspondence approach. We also showed that correcting OCR driven errors helps to improve the performance of the proposed methods.

- **Explaining/Detecting Similarities between Entities from Different Domains**
  Motivated by the lack of evidences to support the similarity computation in vector spaces, we have introduced a novel problem of explaining the similarity of terms by finding their commonalities and aligned differences. In particular, we have proposed two unsupervised methods to solve this task and we have successfully demonstrated their effectiveness in the case of entities from heterogeneous spaces. We first defined several criteria to extract effective evidences to support similarity justification between any two terms. We then proposed two approaches for selecting explanatory terms based on the local and global characteristics. We next evaluated the proposed methods on the New York Times Archive and we demonstrated that capturing both local (quality-based method) and global (systematicity-based method) characteristics among the aspects of the two entities can successfully detect similarity of compared entities. We also introduced two views for result investigation to better present the results discovered in the above two topics.

- **Detecting Cause-Effect Relationships in Text Archives**
  We have investigated how the objects change and what are the effects of these changes by detecting causal relations in a specific type of text archive, online product reviews. We are particularly interested in understanding social impact of technology and in discovering how changes of product features influence changes in our social lives. The intriguing characteristics of our research is a novel approach in which we attempt at automatically estimating the effects of technology and product evolution on our lives. For this we specifically proposed detecting cause-effect relations on word time series. We used not only the frequency-based time series but also proposed constructing time series that capture changes in word usage over time by applying neural networks. Both temporal representations provided complementary results as demonstrated in the experiments. Furthermore, we aggregated the binary causal relations either by meaning or by co-causal relations for obtaining composite causal patterns. Experimental evaluation demonstrated that the proposed composite causality detection methods outperform baselines and binary causality detection methods both when compared with ground truth as well as when evaluated by human annotators.

Finally, technical and social contributions of these researches are summarized as follows:

Technical Contributions

- We proposed a series of methods to bridge the gap between different domains, supporting the object search, entity comparison as well as the analysis of text archives.
- We proposed several methodologies to approach the research task of causality detection, helping users to better understand the changes occurred in text archives.

Social Contributions

- Thanks to our contributions, we have conquered several search problems and lowered the accessibility barriers for average users to explore and understand text archive data.
- Our methodologies can aid with education objectives to let young generations learn more about knowledge unknown to them such as knowledge about the past.
- Our research works provide computational support for sociologist and historians who spend much effort in summarizing the history and analyzing the impact of technology on the society, enabling to construct machines which can automatically examine historical data and generate easy-to-understand results.

6.2 Future Directions

Several promising research directions appeared thanks to this work that can be further explored in future work.

- ***Spatial Transformation.*** In the current work, we focused on solving terminology gap problem between different time periods. In the future, we plan to extend the current approach of temporal transformation to *spatial transformation* in order to support searching for counterparts in unknown locations (e.g., in a different country). For example, contents in a document archive can be categorized by country where each document is judged as being related to a particular country. Then our methodology can be adapted to construct transformations across different countries. Spatial transformation can be applied to the scenarios in which users search for the "similar" term or entity (e.g., a restaurant, particular custom, sightseeing spot, peculiar building) in another country than the country of their location (e.g., searching "from" Japan "to" USA).
- ***Analogy-based Search.*** Current search engines are predominantly based on key word matching. However, searching power and flexibility are constrained when users do not know good key words. We plan to implement analogy-based search functionality which can search in the way similar to the "like" operation in SQL language used in relational

databases. This kind of "like" search operator does not rely on usual key word matching, but it would apply the across-domain similarity and refer to the counterparts or equivalent entities of the query.

- ***Detecting Differences between Entities.*** As for the topic of entity comparison, besides similarities, we can extend our approach to detect differences between entities across different domains in order to enrich the comparisons from multiple perspectives.

- ***Extended Causality Detection.*** We believe that the proposed methods for causality detection can be also applied to other scenarios besides online product reviews (e.g., in the collections of news articles or scientific publications) with minor adaptations. We are interested in testing other scenarios in the future. In addition, as mentioned in Chapter 1, the transformation technique can help improve the performance of causality detection.

- ***Summarization of Entity Evolution.*** In the future, we would like to explore the way to overview the relations between entities. For example, the evolutionary chain can help to depict which entity substitutes another one and how they evolve. For each entity in such an evolutionary chain, we can also list the possible effects caused by the occurrence of that entity, giving the user a fuller picture about the evolution of entities along the time.

- ***Visualization and Web Services.*** Finally, we plan to create a demonstration version for the techniques that we developed as a web service for any users to freely query entities they are interested in. We intend to build interactive visualization for easy understanding and interpretation of the results.

BIBLIOGRAPHY

[1] R. P. Abelson and A. Levi. Decision making and decision theory. 1985.

[2] J. Aitchisonn. *Language Change, Progress or Decay?* Cambridge University Press, 2001.

[3] J. Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012.

[4] L. AlSumait, D. Barbará, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *2008 eighth IEEE international conference on data mining*, pages 3–12. IEEE, 2008.

[5] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *In Proc. of SIGKDD*, pages 66–75. ACM, 2007.

[6] F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *Signal Processing, IEEE Transactions on*, 52(8):2189–2199, 2004.

[7] K. Berberich, S. J. Bedathur, M. Sozio, and G. Weikum. Bridging the terminology gap in web archive search. In Proc. of WebDB, 2009.

[8] E. Billauer. PEAKDET. `http://billauer.co.il/peakdet.html/`. [Online; accessed 03-March-2016].

[9] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, 2009.

[10] E. Blanco, N. Castell, and D. I. Moldovan. Causal relation extraction. In *In Proc. of LREC*, 2008.

[11] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaption for sentiment classification. In Proc. of ACL, pages 440–447, 2007.

[12] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In Proc. of EMNLP, pages 120–128, 2006.

[13] E. Borovikov. A survey of modern optical character recognition techniques. *arXiv preprint arXiv:1412.4183*, 2014.

[14] K. H. Brodersen, F. Gallusser, J. Koehler, N. Remy, S. L. Scott, et al. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1):247–274, 2015.

[15] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[16] P. Cai, W. Gao, A. Zhou, and K. Wong. Relevant knowledge helps in choosing right teacher: Active query selection for ranking adaptation. In Proc. of SIGIR, pages 115–124, 2011.

[17] L. Campbell. *Historical Linguistics, 2nd edition*. MIT Press, 2004.

[18] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[19] D.-S. Chang and K.-S. Choi. Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. *Information processing & management*, 42(3):662–678, 2006.

[20] G. A. Cohen. *Karl Marx's theory of history: A defence*. Oxford: Clarendon Press, 2000.

[21] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[22] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. J. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2412–2421, 2011.

[23] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[24] B. Falkenhainer, K. D. Forbus, and D. Gentner. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63, 1989.

[25] L. Ferreira, N. Jakob, and I. Gurevych. A comparative study of feature extraction algorithms in customer reviews. In *In Proc. of Semantic Computing*, pages 144–151. IEEE, 2008.

[26] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[27] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[28] W. Gao, P. Cai, K. Wong, and A. Zhou. Learning to rank only using training data from related domain. In Proc. of SIGIR, pages 162–169, 2010.

[29] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.

[30] D. Gentner and A. B. Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997.

[31] P. D. Gilbert. Combining var estimation and state space model reduction for simple good predictions. *Journal of Forecasting*, 14(3):229–250, 1995.

[32] T. Gilovich. Seeing the past in the present: The effect of associations to familiar events on judgments and decisions. *Journal of Personality and Social Psychology*, 40(5):797, 1981.

[33] R. Girju. Automatic detection of causal relations for question answering. In *In Proc. of ACL workshop*, pages 76–83. Association for Computational Linguistics, 2003.

[34] R. Girju and D. Moldovan. Mining answers for causation questions. In *In Proc. of AAAI symposium on mining answers from texts and knowledge bases*, 2002.

[35] R. Girju, D. I. Moldovan, et al. Text mining for causal relations. In *In Proc. of FLAIRS*, pages 360–364, 2002.

[36] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[37] C. Grover, S. Givon, R. Tobin, and J. Ball. Named entity recognition for digitised historical texts. In *LREC*. Citeseer, 2008.

[38] C. J. Halperin, G. Y. R. J. Loewenberg, P. Kolchin, R. Berthoff, D. Moltke-Hansen, F. McDonald, G. McWhinehy, J. Leopold, A. O. Hill, and J. Boyd H. Hill. Comparative history in theory and practice: A discussion. *The American Historical Review*, 87(1):123–143, 1982.

[39] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal aspects of computing*, 6(5):512–535, 1994.

[40] Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[41] L. P. Hartley. *The Go-Between*. Harmondsworth : Penguin Books in association with Hamish Hamilton, 1958.

[42] J. R. Hobbs. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209, 2005.

[43] M. Hu and B. Liu. Mining and summarizing customer reviews. In *In Proc. of SIGKDD*, pages 168–177. ACM, 2004.

[44] M. Hu and B. Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.

[45] G. Hughes. *Words in Time: A Social History of the English Vocabulary*. Basil Blackwell, 1988.

[46] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.

[47] A. Jatowt and K. Duh. A framework for analyzing semantic change of words across time. In Proc. of JCDL, pages 229–238, 2014.

[48] A. C. Kaluarachchi, A. S. Varde, S. Bedathur, G. Weikum, J. Peng, and A. Feldman. Incorporating terminology evolution for query translation in text retrieval with association rules. In Proc. of CIKM, pages 1789–1792, 2010.

[49] N. Kanhabua and K. Nørvåg. Exploiting time-based synonyms in searching document archives. In Proc. of JCDL, pages 79–88, 2010.

[50] N. Kanhabua and K. Nørvåg. A comparison of time-aware ranking methods. In Proc. of SIGIR, pages 1257–1258, 2011.

[51] R. M. Kaplan and G. Berry-Rogghe. Knowledge-based acquisition of causal relationships in text. *Knowledge Acquisition*, 3(3):317–337, 1991.

[52] M. P. Kato, H. Ohshima, and K. Tanaka. Content-based retrieval for heterogeneous domains: domain adaptation by relative aggregation points. In *In Proc. of SIGIR*, pages 811–820, 2012.

[53] C. S. Khoo, S. Chan, and Y. Niu. Extracting causal knowledge from a medical database using graphical patterns. In *In Proc. of ACL*, pages 336–343. Association for Computational Linguistics, 2000.

[54] E. KıcKıman and M. Richardson. Towards decision support and goal achievement: Identifying action-outcome relationships from social media. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 547–556. ACM, 2015.

[55] R. Killick and I. Eckley. changepoint: An r package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19, 2014.

[56] B. Kim, J. Scott, and S. Kim. Exploring digital libraries through visual interfaces. *In: Kuo Hung Huang: "Digital Libraries - Methods and Applications"*, pages 123–136, 2011.

[57] Y. Kim, Y.-I. Chiu, K. Hanaki, D. Hegde, and S. Petrov. Temporal analysis of language through neural language models. In Proc. of ACL Workshop, pages 61–65, 2014.

[58] S. Kleinberg. *Causality, probability, and time*. Cambridge University Press, 2012.

[59] S. Kleinberg and G. Hripcsak. A review of causal inference for biomedical informatics. *Journal of biomedical informatics*, 44(6):1102–1112, 2011.

[60] S. Kleinberg and B. Mishra. The temporal logic of causal structures. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 303–312. AUAI Press, 2009.

[61] O. Kolomiyets and M.-F. Moens. A survey on question answering technology from an information retrieval perspective. *Inf. Sci.*, 181(24):5412–5434, 2011.

[62] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. In Proc. of WWW, pages 625–635, 2015.

[63] W. Labov. *Principles of Linguistic Change*. Wiley-Blackwell, 2010.

[64] R. J. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977.

[65] E. Lieberman, J. B. Michel, J. Jackson, T. Tang, and M. A. Nowak. Quantifying the evolutionary dynamics of language. *Nature*, pages 713–716, 2007.

[66] P. G. Lindemann and A. B. Markman. Alignability and attribute importance in choice. In *Proceedings of the eighteenth annual meeting of the Cognitive Science Society*, pages 358–363, 1996.

[67] X. Ling, W. Dai, G. R. Xue, Q. Yang, and Y. Yu. Spectral domain-transfer learning. In Proc. of SIGKDD, pages 488–496, 2008.

[68] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[69] S. M. Lucas, A. Tams, S. J. Cho, S. Ryu, and A. Downton. Robust word recognition for museum archive card indexing. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 144–148. IEEE, 2001.

[70] A. Mazeika, T. Tylenda, and G. Weikum. Entity timelines: Visual analytics and named entity evolution. In *In Proc. of CIKM*, pages 2585–2588. ACM, 2011.

[71] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[72] R. Mihalcea and V. Nastase. Word epoch disambiguation: Finding how words change over time. In Proc. of ACL, pages 259–263, 2012.

[73] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In Proc. of EMNLP, pages 404–411, 2004.

[74] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Proc. of ICLR Workshop, 2013.

[75] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[76] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representation of phrases and their compositionality. In Proc. of NIPS, pages 3111–3119, 2013.

[77] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named entity extraction from noisy input: speech and ocr. In *Proceedings of the sixth conference on Applied natural language processing*, pages 316–324. Association for Computational Linguistics, 2000.

[78] P. Mirza. Extracting temporal and causal relations between events. In *ACL (Student Research Workshop)*, pages 10–17, 2014.

[79] K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[80] P. Nation and R. Waring. Vocabulary size, text coverage and word lists. *Vocabulary: Description, acquisition and pedagogy*, 14:6–19, 1997.

[81] P. Norvig. Natural language corpus data. *Beautiful Data*, pages 219–242, 2009.

[82] H. Ohshima and K. Tanaka. High-speed detection of ontological knowledge and bi-directional lexico-syntactic patterns from the web. *Journal of Software*, 5(2):195–205, 2010.

[83] S. Pan and Q. Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):713–716, 2010.

[84] M. Pargel, Q. D. Atkinson, and A. Meade. Frequency of word-use predicts rates of lexical evolution throughout indo-european history. *Nature*, 449:717–720, 2007.

[85] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *In Proc. of HLT-NAACL*, pages 38–41. Association for Computational Linguistics, 2004.

[86] M. Piotrowski. Natural language processing for historical texts. *Synthesis Lectures on Human Language Technologies*, 5(2):1–157, 2012.

[87] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[88] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st International Conference on World Wide Web*, In Proc. of WWW, pages 909–918, 2012.

[89] Resources:. The history of portable audio. `http://www.ehow.com/about_5292437_history-portable-audio.html`, 2001. Accessed: 2016-01-29.

[90] Resources:. External flash. `http://photographycourse.net/lessons/external-flash/`, 2008. Accessed: 2016-01-29.

[91] Resources:. Digital camera advantages. `http://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/knowhow25.html`, 2009. Accessed: 2016-01-29.

[92] Resources:. The history of car radios. `http://www.caranddriver.com/features/the-history-of-car-radios`, 2010. Accessed: 2016-01-29.

[93] Resources:. Benefits and limitations of dslrs vs. camcorders. `http://www.videomaker.com/videonews/2012/07/benefits-and-limitations-of-dslrs-vs-camcorders/`, 2012. Accessed: 2016-01-29.

[94] Resources:. The disadvantages of film cameras. http://www.ehow.com/info_8078035_disadvantages-film-cameras.html, 2012. Accessed: 2016-01-29.

[95] Resources:. Top 10 historical music players. http://cassette-to-mp3-review.toptenreviews.com/top-10-historical-music-players.htmls, 2013. Accessed: 2016-01-29.

[96] Resources:. 10 things you need to know about camera lenses. http://www.ebay.com/gds/10-Things-You-Need-to-Know-About-Camera-Lenses-/10000000177628167/g.html, 2014. Accessed: 2016-01-29.

[97] Resources:. Camera innovation: 10 products that are changing how we take photos and videos. http://www.techrepublic.com/article/camera-innovation-10-products-that-are-changing-how-we-take-photos-and-videos, 2014. Accessed: 2016-01-29.

[98] Resources:. A complete history of portable music players. http://www.ebay.com/gds/A-Complete-History-of-Portable-Music-Players-/10000000177628958/g.html, 2014. Accessed: 2016-01-29.

[99] Resources:. Thank you for the music: A potted pictorial history of portable music devices. http://www.telegraph.co.uk/news/picturegalleries/uknews/10938261/Thank-you-for-the-music-A-potted-pictorial-history-of-portable-music-devices.html, 2014. Accessed: 2016-01-29.

[100] Resources:. Top five mp3 players for running. http://aminebombom.hubpages.com/hub/top-5-mp3-players-for-running, 2015. Accessed: 2016-01-29.

[101] M. W. Reynaert. Character confusion versus focus word-based correction of spelling and ocr variants in corpora. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(2):173–187, 2011.

[102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ, San Diego La Jolla Inst. For Cognitive Science, 1985.

[103] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[104] E. Sandhaus. The new york times annotated corpus overview. *The New York Times Company, Research & Develop.*, pages 1–22, 2008.

[105] V. Singh and S. K. Dwivedi. Question answering: A survey of research, techniques and issues. *International Journal of Information Retrieval Research*, 4(3):14–33, 2014.

[106] J. K. Sparck and V. C. Rijsbergen. *Report on the need for and provision of an ideal information retrieval test collection*. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[107] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

[108] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, In Proc. of WWW, pages 697–706, 2007.

[109] N. Tahmasebi, G. Gossen, N. Kanhabua, H. Holzmann, and T. Risse. Neer: An unsupervised method for named entity evolution recognition. In Proc. of Coling, pages 2553–2568, 2012.

[110] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *In Proc. of ACL*, pages 417–424. Association for Computational Linguistics, 2002.

[111] P. D. Turney. Expressing implicit semantic relations without supervision. *CoRR*, 2006.

[112] P. D. Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, pages 615–655, 2008.

[113] L. Von Ahn. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 418–419. IEEE, 2009.

[114] D. Wang, S. Zhu, T. Li, and Y. Gong. Comparative document summarization via discriminative sentence selection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(3):12, 2012.

[115] H. Wang, H. Huang, F. Nie, and C. Ding. Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization. In Proc. of SIGIR, pages 933–942, 2011.

[116] M. L. Wilson, M. C. schraefel, and R. W. White. Evaluating advanced search interfaces using established information-seeking models. *J. Am. Soc. Inf. Sci. Technol.*, 60(7):1407–1422, 2009.

[117] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634. ACM, 2008.

[118] J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *In Proc. of SIGKDD*, pages 1079–1088. ACM, 2010.

[119] Y. Zhang, A. Jatowt, S. S. Bhowmick, and K. Tanaka. Omnia mutantur, nihil interit: Connecting past with present by finding corresponding terms across time. In Proc. of ACL, pages 645–655, 2015.

PUBLICATIONS

International Journal Papers

1. Yating Zhang, Adam Jatowt, Katsumi Tanaka. Causal Relationship Detection in Archival Collections of Product Reviews for Understanding Technology Evolution. The Journal of ACM Transaction on Information Systems **(ACM TOIS *journal*)**, 2016. (In Press)

2. Yating Zhang, Adam Jatowt, Sourav S. Bhowmick, Katsumi Tanaka. The Past is Not a Foreign Country: Detecting Semantically Similar Terms across Time. The Journal IEEE Transactions on Knowledge and Data Engineering **(IEEE TKDE *journal*)**, 2016. (In Press)

International Conference Papers

1. Yating Zhang, Adam Jatowt, Sourav S. Bhowmick, Katsumi Tanaka. Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time. In the 53rd Annual Meeting of the Association for Computational Linguistics in 2015 **(ACL 2015)**, pp. 645-655, 2015.

2. Yating Zhang, Adam Jatowt, Katsumi Tanaka: Detecting Evolution of Concepts based on Cause-Effect Relationships in Online Reviews. In Proceedings of the 25th International World Wide Web Conference **(WWW 2016)**, ACM Press, pp. 649-660, 2016.

3. Yating Zhang, Adam Jatowt, Katsumi Tanaka: How Good are Word Embeddings? Automatically Explaining Similarity of Terms. (Submitted)

Domestic Conferences/Workshops

1. Yating Zhang, Adam Jatowt, Katsumi Tanaka. Search for Images of Historical Objects using Wikipedia. The 6th Forum on Data Engineering and Information Management **(DEIM 2014)**, A1-2, 2014. ("The Best Student Presentation Award")

2. Yating Zhang, Adam Jatowt, Katsumi Tanaka. Bridging the Gap between the Past and the Present: Finding Corresponding Objects across Time. The 7th Forum on Data Engineering

and Information Management **(DEIM 2015)**, A3-4, 2015. ("The Best Student Presentation Award")

3. Yating Zhang, Adam Jatowt, Katsumi Tanaka. Finding "Similar" Concepts with Evidences Across Different Feature Vector Spaces. The 8th Forum on Data Engineering and Information Management **(DEIM 2016)**, A3-3, 2016. ("The Best Student Presentation Award")

4. Yating Zhang, Adam Jatowt, Katsumi Tanaka. Towards Mining Object Evolution from the Web. The 6th International Workshop with Mentors on Databases, Web and Information Management for Young Researchers, 2014 年 8 月.