Build a Causality Database from Yelp Reviews

Xing Shi
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
xingshi@usc.edu

Ai He
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
aihe@usc.edu

Abstract

In this work, we have built a database containing *Reason-Consequence* pairs extracted from a Yelp.com review dataset. The difference between our work and the existing causality database lies in that the reason and consequence of a pair are sentences-level instead of word level. Besides the pain text version of *Reason-Consequence* pairs, we also provided the structured version, which contains the subject, verb, direct object, indirect object and their relevant adjectives, as well as the reference chain which appears between the reason and the consequence of a certain pair. Our precision of causality relation is relatively high, achieving 84%, making it a good resource to analysis the causality relation as well as users' opinion.

1 Introduction

We use the following example to describe the problem we want to solve:

Example:

*Economic Optimism: Ezra Klein points out the **reasons** for economic optimism. 1) Health-care costs are slowing, and not just **because** of the recession. If the cost controls being rolled out in Obamacare can hold the trend, much in the U.S. economy, from the budget deficit to wages, looks much brighter. 2) Housing is turning around. And **if** housing turns, **then** the Federal Reserves super-low interest rates could really begin to drive a recovery.*

From the above paragraph, we want to extract the following facts:

- #1:(Reason:*Health-care costs are slowing*; Consequence: *economic optimism*.)
- #2:(Reason: *Housing is turn around*; Consequence: *economic optimism*.)
- #3:(Reason: *the recession*; Consequence: *Health-care costs are slowing*.)

We want to build a knowledge base where each entry is a reason-consequence pair having the following format:

- Entry:(ReasonFact1; Consequence: Fact2)
- Fact: [Compound noun— simple phrase]
- Simple phrase: (subject; verb; direct object; indirect object)

In the above example, there could be another reason-consequence pair.

- #4(Reason: *housing turns*; Consequence: *the Federal Reserves super-low interest rates could really begin to drive a recovery*.)

However, the consequence part is too complicated, we do not plan to analysis it.

In summary, our goal is to build a database which contains the *Reason-Consequence* pairs, and all these pairs are sentence-level. We have primary two motivations to build this database:

1. Use this dataset as a resource to build a causality detecting classifier.
2. Use this dataset to analysis the users' deep opinion.

The first motivation requires our dataset to have the following 2 aspects:

1. High Precision. As we want to use this dataset as a kind of labeled data, so we require all the entries are as correct as possible.
2. Structured. A structured dataset is much easier to be used by other application than the plain text. So in our project, we tried two means to make the data entries structured.

First is to extract the subject, verb, direct object, indirect object and their relevant adjectives. Second, extract the co-reference chain between the reason and consequence in a certain pair.

2 Methodology

2.1 System Overview

In this section, I will discuss our methodology to extract the reason-consequence pairs, and some opinion mining work we have done.

The figure 1 describe the pipeline of the whole framework. First, we filtered all the reviews text from yelp dataset, and push these reviews into our data preprocessing component. This component contains 4 part: Sentence Segmenter, Discourse Unit Segmenter, Reference Resolution, Word Stemming. One of the aims of data preprocessing is to split the whole text into appreciated text spans, which are the basic building block of causality relation extraction. Another aim is to prepare the data for tuple extracted to make the pairs structured.

After preprocessing, we can enter into the "Pattern Based Extraction" phase, where we can use some connectives patterns such as "because","since" and so on to extract some rough reason-consequence pairs. When all pairs are extracted, we start to extract the tuples, i.e. extract the subject, verb, direct object, indirect object and their relevant adjectives. Till now, our database is built with two version of pairs, those of plain text and those in tuple format.

One of our motivation is to do opinion mining from the causality database, so we use a clustering algorithm to analysis both the consequences and the reasons, trying to discover some useful and insightful information.

2.2 Data preprocessing

In this section, we will introduce all four parts of data preprocessing in two aspects: their function and the relevant tools.

1. Sentence Segmenter: Each review is a collection of sentences. Here we applied the sentence segmenter in nltk (Bird et al. [2009]) of latest version on the raw review to split the whole paragraph into sentences.

2. Discourse Unit Segmenter: Here we followed the definition of elementary discourse unit(EDU) proposed by Marcu [1998]. The EDU is a clause or clause-like sentence structure. We applied the SPADE discourse parser (Soricut and Marcu [2003]) on each sentence of the review to generate a set of EDUs(or clauses).

3. Reference Resolution: Our goal is to build a knowledge base containing the causality relations between two phrases, which should not contains pronouns like "it", "he", "she", in order to avoid ambiguation. In our system, we applied Stanford Core NLP tools (Raghunathan et al. [2010],Kozareva [2012],De Marneffe et al. [2006],Toutanova and Manning [2000],Toutanova et al. [2003],Finkel et al. [2005],Lee et al. [2013],Lee et al. [2011]) to do this task.

4. Word Stemming: Because of the same reason (avoid ambiguation), we do not want the different forms of the same word to affect the result. And we also utilize NLTK package.

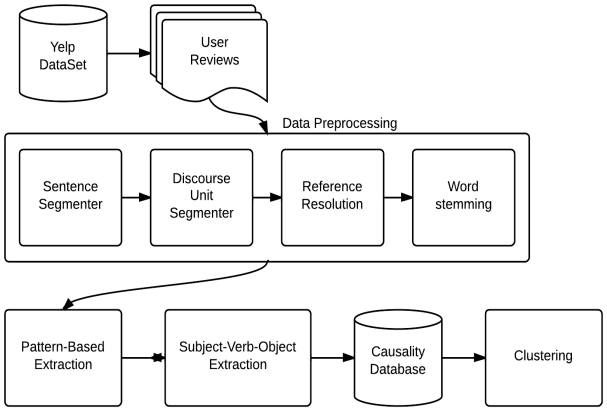

Figure 1: The overview of framework design

2.3 Pattern-based Extraction

In order to get the relatively high precision result and apply an unsupervised method to extract the causality relations, we decide to use the pattern-based extraction method. The discourse information is mainly conveyed by the connectives as well as those words at the edge of clauses. So that our patterns considers using the features of the connectives and the "edge" features of clauses.

After the data preprocessing, every review in our data is a sequence of elementary discourse

units. So our pattern is also a sequence pattern. First, we came up several handcraft trigger words that indicate the causality relations, shown as following:

- so
- in order to
- thus
- because
- ...

Let's take "because" as an example, the EDU that serves as reason is easy to find, just the same EDU with the word "because". The problem is where is the EDU that contains the relevant consequence. According to Prasad et al. [2010], the Arg1 (In the Penn Discourse Treebank, Arg1 is similar to the EDU that contians the consequence here.) is most likely to be located in the immediate previous sentences of the sentence containing the connectives.Actually, this is the strong baseline of Prasad et al. [2010], and the likelihood of such situation has achieved 83%. So we simply make a decision that the consequence is located in the immediate previous EDU of the word "because".

So we have our first pattern, "C R$\hat{b}$ecause". This pattern means that the reason is located in the EDU which starts with word "because", and the relative consequence is located in the previous EDU. Then we tried to replace the word "because" to other words in our handcraft list to find the best one. The experiment shows that the word "because" achieve the best precision, where as other words had a lot of noises.

After considering more "edge" features, we come up with our final two patterns, as shown in figure 2:

1. Single Reason Pattern:
   "C+ R∧because_/p". This means that the reason is located in an EDU that begins with word "because" and ends with punctuations.The consequence is in the previous EDU of which the first letter must be initialized.

2. Double Reason Pattern:
   "C+ R∧because<> . R∧and_/p". This pattern means the one reason is located in an EDU that begins with word "because" and does not end with full stop, the other reason is in the next EDU which begins with word "and" and ends with punctuation. The relevant consequence is in the previous EDU of which the first letter must be initialized.

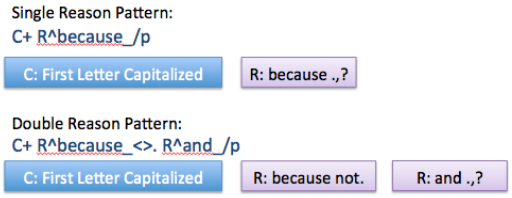

Figure 2: Sequence Patterns

Based on only these two patterns, we have achieve a relatively high precision. The figure 3 shows the distribution of results extracted by the two patterns based on 172 samples. The single reason pairs are about 86% whereas the double reason pairs cover the rest 14%.

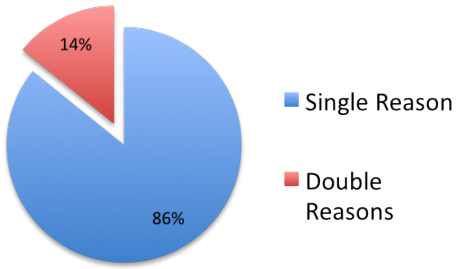

Figure 3: Distribution of single and double reason pairs

2.4 Tuple Extraction

After the previous step, we've got the reason-consequence pairs in plain text. In order to make these pairs structured, we decide to extract the subject, verb, direct object, indirect object and their relevant adjectives of both consequence and reason EDU. So we design a tuple shown as follows, and try to extract the relevant part to fill the blanks of the tuple.

```
tuple:
{
'subj':None,'subj_adj':None,
'verb':None,'adv':None,'neg':False,
'dobj':None,'dobj_adj':None,
'iobj':None,'iobj_adj':None
}
```

We first utilized the Stanford Dependency Parser (De Marneffe et al. [2006]) to get the syn-

tactic information of each clause, then extracting the relevant part is just a simple matching task.

2.5 Clustering

We did clustering in both consequences and corresponding reasons clauses using LSA algorithm. The number of consequences clauses to be clustered is 10545 and we used 20 as the clustering number. We didn't use stopwords here because all the consequences clauses are quite short and the same reason to set minimal frequency of words to be 1.

There is one particular cluster in the result where most consequences clauses are about customer giving stars. Then we partitioned them into 5 parts(1-5 stars). Now for each consequence clause in each category of star, its corresponding reasons clauses can be found. In these clauses, we did another clustering algorithm to see what are the reasons for each category of star.

3 Experiment

3.1 Dataset

We got the dataset from Yelp, the Yelp Dataset Challenge(`http://www.yelp.com/dataset_challenge/`). The data is a generous sample of yelp's data from the greater Phoenix, AZ metropolitan area. It contains:

- 11,537 businesses
- 8,282 checkin sets
- 43,873 users
- 229,907 reviews

Among which, we are interested in the large-scale of user generated reviews. The data is provided in JSON format, the following is the detailed format of reviews dataset:

```
review
{
  'type': 'review',
  'business_id':
  (encrypted business id),
  'user_id': (encrypted user id),
  'stars':
  (star rating, rounded to half-stars),
  'text': (review text),
  'date':
  (date, formatted like '2012-03-14'),
  'votes': {(vote type): (count)}
}
```

3.2 Overall Performance

From the 229,907 reviews, we extracted 11,272 reason-consequence pairs. The extraction rate is about 5%. In our project, as one of our motivation is to use this database as a resource to build other application, so we hope that the precision would be high. The figure 4 reflects our overall performance. In order to evaluate the precision, we labeled 200 samples. The overall precision over these 200 samples achieves 84%, among which the double reason pairs is much higher, approaching 93% compared to 83% of the single pair reasons.

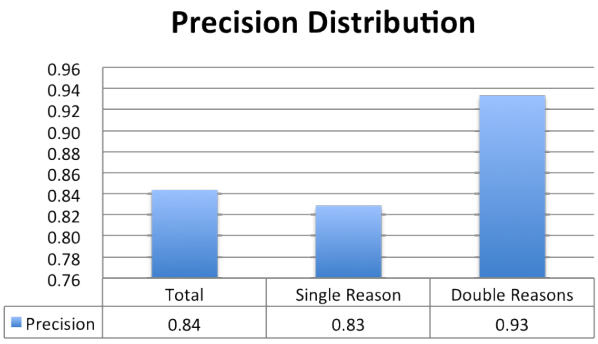

Figure 4: Precision Distribution

3.3 Semantic Analysis

In this section, we would like to analysis the causality relations from both semantic and statistic point of views. Here we give two examples from our data sets:

> Reason#1: because the food is good , fresh , fast
> Reason#2: and the service is friendly and quick .
> Consequence: I give it four stars

and

> Reason: because I have seen better.
> Consequence:Only 4 stars

From the above two examples, we can easily find that the words in reasons, like "food","fast","friendly" have little semantic relations with the words in consequences, like "give","stars". So we did an evaluation based on 172 samples, whose task is to label whether there exist semantic similar words between the reason and the consequence. Figure 5 show that only 8

percent of those pairs contains semantic related words. This may indicate that discourse parsing, or at least causality detecting is not a semantic problem.

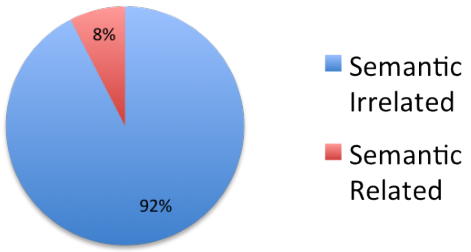

Figure 5: Pie chart of semantic related situation

But this doesn't mean that we can not find some cues to detect the discourse relations from the content feature. We decide to do clustering over all the consequences, and found a cluster of which all consequences express how many stars the custom give the business. Further, we analysis the reasons of this kind of consequences, the results show in figure 6 and 7

```
3 stars
a).service:
1.because the service is 4 star
2.because 1 ) They had good service , 2 ) While they do
n't cook the greatest steak in the work , their Rib-eye
was n't bad , 3 ) The side of mash potatoes were fairly
tasty .
3.because service is a big chaos ! .


b).food:
1.because everything else about the food was all right ,
2.because while the food is good ,
3.because the beer is good ,
4.because their food is quite good .
5.because it is good , not great .
6.because the food is good .
7.because one night I got a really bad hot dog .
```

Figure 6: the reasons of reviews giving 3 stars

Although these are some representative results, we can find that the reasons are actually distributed in a narrow scope: the reasons affecting people's view are almost just service, food and price. So we may draw a conclusion: the causality realtion detecting is more a statistic problem than a semantic problem, and the statistic only have effects under a specific domain.

3.4 Causality Complexity Analysis

We hope that our database is structured and thus can be easily used by others. So we extracted the tuples from each EDU expecting that this tuples can convey most of the causality relations. We

```
4 stars:
a).food:
1.and food was hot 2 ) The pizza 's all looked
consistent 3 ) The ingredients were fresh 4 ) good
flavor 5 ) Filling !
2.because the food is good , fresh , fast
3.because while it was good ,
4.because I love the food .
5.because the food was really good and the service .

b).price:
1.because the programming is SO expensive
2.and it is a bit expensive .
3.because it is a bit pricy ,
4.because for a walk up and order spot , this place is
pretty pricey .


c).service:
1.Because the service can be absolutely abysmal at times
.
2.because the service can sometimes be a little off
3.because the service was terrible .
4.because customer service could be a tad better .
```

Figure 7: the reasons of reviews giving 4 starts

did an evaluation based 100 examples, shown in figure 8. The result shows that only 36% tuple pairs can convey the complete causality information, which is disappointing. In the rest 64% pairs, either the tuple itself is not extracted correctly, nor the causality relation lies on other part of the sentence.

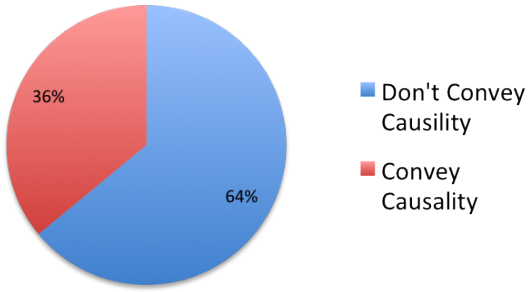

Figure 8: Pie chart of ratio conveying causality

3.5 Clustering Result Analysis

In the previous, we think user review contains a lot of noises and people's utterances are really various. However, from a result of consequences clauses clustering, we found some clustered classes were amazing, shown in figure 9. From the clustering result, the customers tend to use the same way to express their opinions. Totally, the number of consequences clauses in this cluster is 364. Then for the number of reasons clauses that correspond to consequences clauses in each kind of star, 1 star has 2 reasons, 2 has 92, 3 has 123, 4 has 123 and 5 has 84. In the clustering result of each reasons group above, we found

some classes mainly mention one specific aspect, such as food, service or price, shown in figure 6 and 7.

```
<doc rank="193229_0">I give Posh two stars</doc>
<doc rank="109991_1">I gave this place five stars</doc>
<doc rank="84705_3">I gave them three stars</doc>
<doc rank="169272_14">I gave it three stars</doc>
<doc rank="119002_11">I 'm giving this Morton 's 4 stars</doc>
<doc rank="52368_0">I give this four stars</doc>
<doc rank="191849_0">I give rice paper five stars</doc>
<doc rank="210292_0">Starting off with three stars for Grazie</doc>
<doc rank="66203_2">I give it 5 stars</doc>
<doc rank="80733_4">I 'm still giving 4 stars</doc>
<doc rank="20529_15">Oh and five stars</doc>
<doc rank="106029_8">Perhaps people give Zen 32 4 or 5 stars</doc>
<doc rank="195676_26">I give them 4 stars</doc>
<doc rank="139001_4">This location only gets four stars</doc>
<doc rank="153761_5">Two stars</doc>
<doc rank="164620_55">Salt Cellar only gets two stars</doc>
<doc rank="178778_3">I only gave it 3 stars</doc>
<doc rank="127142_22">I gave this business two stars</doc>
<doc rank="177929_8">Three stars</doc>
<doc rank="135767_0">I was going to give it 2 stars</doc>
<doc rank="181361_0">I am giving this Restaurant four stars</doc>
<doc rank="166705_22">I give this two stars</doc>
<doc rank="198479_4">So why the five stars ?</doc>
<doc rank="1906_18">I give it three stars</doc>
```

Figure 9: Consequences in one cluster

4 Conclusion

In our project, we built a database contains sentence-level reason-consequence pairs with high precision. Also, we tried to make these pairs structured by extracting the subject, verb, direct object, indirect object and their relevant adjectives. We also found some interesting phenomenon: First, the causality detecting problem is more a statistic problem than a semantic problem; Second, the causality information is conveyed by various parts of a sentence; Third, the utterance of people's review is more monotonous that what we expected before.

In the future, we plan to do the following work:

We now have 10,000 more pairs of causality relationship and they have a very high precision which can be treated as the labeled data. We intend to build a model to detect the causality relationship between two clauses and train the model in these 10,000 more pairs.

Also,we plan to use the result of co-relation reference dissolution more deeply. The way we expect to do is to find the common part(the parts that are in the same co-reference chain) and then extract abstract templates from those reasons and consequences pairs. We could build another model or do some casual extraction based these abstract causality pairs.

References

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. O'Reilly Media, 2009.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

Zornitsa Kozareva. Cause-effect relation learning. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, pages 39–43. Association for Computational Linguistics, 2012.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, (Just Accepted):1–54, 2013.

Daniel Marcu. *The rhetorical parsing, summarization, and generation of natural language texts*. PhD thesis, University of Toronto, 1998.

Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Exploiting scope for shallow discourse parsing. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2076–2083, 2010.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics, 2010.

Radu Soricut and Daniel Marcu. Sentence level

discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics, 2003.

Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.