A Dependency Relation-based Method to Identify Attributive Relations and Its Application in Text Summarization

Shamima Mithun and Leila Kosseim

Concordia University
Department of Computer Science and Software Engineering
Montreal, Quebec, Canada
{s_mithun, kosseim}@encs.concordia.ca

Abstract. In this paper, we propose a domain and genre-independent approach to identify the discourse relation called *attributive*, included in Grimes' relation list [7]. An attributive relation provides details about an entity or an event or can be used to illustrate a particular feature about a concept or an entity. Since attributive relations describe attributes or features of an object or an event, they are often used in text summarization (e.g. [2]) and question answering systems (e.g. [12]). However, to our knowledge, no previous work has focused on tagging *attributive* relations automatically. We propose an automatic domain and genre-independent approach to tag attributive relations by utilizing dependency relations of words based on dependency grammars [3]. In this paper, we also show how attributive relations can be utilized in text summarization. By using a subset of the BLOG06[1] corpus, we have evaluated the accuracy of our attributive classifier and compared it to a baseline and human performance using precision, recall, and F-Measure. The evaluation results show that our approach compares favorably with human performance.

1 Introduction

According to [15], "Discourse relations - relations that hold together different parts (i.e. proposition, sentence, or paragraph) of the discourse - are partly responsible for the perceived coherence of a text". In a discourse, different kinds of relations such as *contrast*, *causality* or *elaboration* may be expressed. For example, in the sentence *"If you want the full Vista experience, you'll want a heavy system and graphics hardware, and lots of memory"*, the first and second clauses are related through the discourse relation *condition*. The use of discourse relations have been found useful in many applications such as document summarization (e.g. [1, 2, 13]) and question answering (e.g. [10, 12]). However, these relations are often not considered in computational language applications because domain and genre-independent robust discourse parsers are very few.

[1] http://ir.dcs.gla.ac.uk/test_collections/blog06info.html

In this paper, we propose a domain and genre-independent approach to identify the discourse relation called *attributive*, included in Grimes' relation list [7]. An attributive relation provides details about an entity or an event. For example, in *Mary has a pink coat.*, the sentence exhibits an attributive relation because it provides details about the entity *coat*. Attributive relations can also be used to illustrate a particular feature about a concept or an entity - e.g. *Picasa makes sure your pictures are always organized.* The sentence of this example also contains an attributive relation since it is describing a particular feature of the entity *Picasa*. Even though attributive relations are often used in summarization (e.g. [13]) and question answering systems (e.g. [12]), to our knowledge, no previous work has focused on tagging *attributive* relations automatically. We propose an automatic domain and genre-independent approach to identify whether a sentence contains an attributive relation by utilizing dependency relations of words based on dependency grammars [3]. In this paper, we also show how attributive relations can be utilized in text summarization and how our tagger has been evaluated in that context.

2 Related Work

Currently, to identify discourse relations automatically from multi-documents, only a few approaches are available. The most notable ones are the SPADE parser [14], Jindal et al.'s approach [8], and HILDA [6].

The SPADE parser [14] was developed within the framework of RST (Rhetorical Structure Theory). The SPADE parser identifies discourse relations within a sentence by first identifying elementary discourse units (EDU)s, then identifying discourse relations between two EDUs (clauses) by following the RST theory. However, the attributive relation is not included within these relations.

Another discourse parser is presented in [8]. This parser focuses on tagging the comparison relation. In order to label a clause as containing a *comparison* relation, [8] used a set of keywords and annotated texts, and generate patterns for comparison sentence mining. A Naïve Bayes classifier is then used using the patterns as features to learn a 2-class classifier (comparison and non-comparison). This approach is used in our summarization system (Section 4.2) to tag intra-clausal comparison relations; but again, it does not deal with attributive relations.

Another notable work is that of [6] who designed the discourse parser called HILDA[2] (HIgh-Level Discourse Analyzer) which can tag discourse relations at the text level. First, this parser extracts different lexical and syntactical features from the input texts. Then the parser is trained using the RST Discourse Treebank[3] (RST-DT) corpus. This parser consists of two SVM classifiers. The first classifier finds the most appropriate relation between two textual units and the second classifier verifies whether two adjacent text units should be merged to

[2] HILDA: http://nlp.prendingerlab.net/hilda

[3] http://www.isi.edu/ marcu/discourse/Corpora.html

form a new subtree. However, the source of the parser is not publicly available and again does not tag attributive relations.

Other notable works on discourse parsing and discourse segmentation are proposed by (e.g. [11, 16]). However, the attributive relation is not tagged by any of these approaches. Discourse parsing systems are being developed in other languages than English such as [4] for Spanish.

3 A Method based on Dependency Relations

According to [12], an attributive relation provides details about an entity or event. It can be used to illustrate a particular attribute or feature about a concept or an entity. For example, *Subway sells custom sandwiches and salads.* - contains an attributive relation since it provides an attribute about *Subway.* This relation has been used successfully by [12] in question answering and natural language generation. However, currently, no automatic approach is available to identify attributive relations.

To develop our method to identify attributive relations, we have performed a corpus analysis of 200 attributive sentences from the BLOG06 corpus[4].

A first analysis of our development set showed that 83% of the time, attributive relations occur within a clause; as opposed to many other discourse relations that span across clauses. Due to this, our approach is based on the analysis of single clauses. To identify attributive relations automatically, similarly to Fei et al.'s work [5], we have used dependency relations of words based on dependency grammars [3].

Table 1. Sample Dependency Relations between Words (taken from [5])

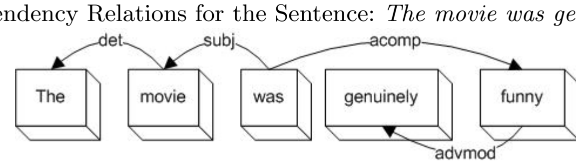
Relation Name	Description	Examples	Parent	Child
subj	subject	I will go	go	I
obj	object	tell her	tell	her
mod	modifier	a nice story	story	nice

Dependency relations of words are defined based on dependency grammars [3]. They refer to the binary relations between two words where one word is the parent (or head) and the other word is the child (or modifier). In this representation, one word can be associated with only one parent but with many children (one word can modify only one other word, but a word can have several modifiers). Therefore, when the dependency relations of a sentence is created it will be in the form of a tree (called a dependency tree). Typical dependency relations are shown in Table 1.

[4] BLOG06 is a TREC test collection, created and distributed by the University of Glasgow to support research on information retrieval and related technologies. BLOG06 consists of 100,649 blogs which were collected over an 11 week period (a total of 77 days) from late 2005 and early 2006. The total size of collection is 25 gigabytes. In this corpus, blogs vary significantly in size, ranging from 44 words to 3000 words.

Different words of a sentence can be related using dependency relations directly or based on the transitivity of these relations. For example, the dependency relations of the sentence *"The movie was genuinely funny."* as produced by the Stanford parser[5] is shown in Figure 1.

Fig. 1. Dependency Relations for the Sentence: *The movie was genuinely funny.*

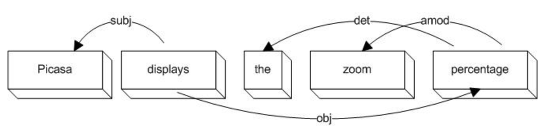


The head of the arrow points to the child, the tail comes from the parent, and the tag on the arrow indicates the dependency relation type. For example, in Figure 1, both words *movie* and *funny* are modifiers of the word *was*. While, the word *movie* is the subject of the word *was*, the word *funny* is a direct adjectival complement (`acomp`) to the word *was*. With the help of dependency relations, it is possible to find how different words of a sentence are related.

In order to develop our classifier, we have first parsed the sentences of our development set using the Stanford parser. A manual analysis of these parses showed that to be classified as an *attributive* sentence, the topic of the sentence needs to be the descendant of a verb and be in a subject or object relation with it. However, the topic and the verb can be related in several ways; which we describe by 3 heuristic rules:

Heuristic 1: The Topic is a Direct Nominal Subject: The *topic* is a direct nominal subject, a noun phrase that is the syntactic subject of the *verb* (e.g., `subj` in the Stanford parser).

Fig. 2. Example of Heuristic 1 to Tag the Attributive Relation



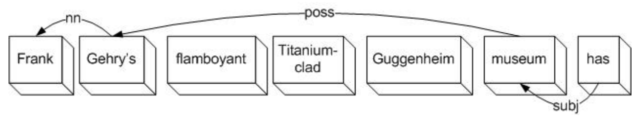
For example, the sentence *"Picasa displays the zoom percentage"* contains an attributive relation where the topic "*Picasa*" is directly related to the verb "*displays*" using the dependency relation `subj` (shown in Figure 2). This is the most frequently encountered dependency relation which occurs within a clause in our attributive development set and accounts for 42% of the development set.

Heuristic 2: A Noun is the Syntactic Subject and the Topic is a Modifier of the Noun: A noun is the syntactic subject of the sentence and the *topic* is a modifier of the noun. This heuristic rule accounts for modifiers that can be a noun compound modifier (e.g., `nn` in the Stanford parser),

[5] http://nlp.stanford.edu/software/lex-parser.shtml

a propositional modifier (e.g., `prep` in the Stanford parser) or a possession modifier (e.g., `poss` in the Stanford parser).

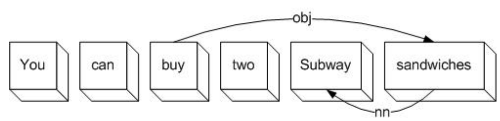
Fig. 3. Example of Heuristic 2 to Tag the Attributive Relation



For example, the sentence *"Frank Gehry's flamboyant, titanium-clad Guggenheim Museum has a similar relationship to the old, masonry city around it."* contains an attributive relation where the noun "*Museum*" is the subject of the sentence and the topic "*Frank Gehry*" is a possession modifier of the noun "*Museum*" (a partial dependency tree is shown in Figure 3). These dependency relations account for 38% of the development set.

Heuristic 3: A Noun is the Syntactic Direct Object and the Topic is a Modifier of the Noun: A noun is the syntactic direct object of the *verb* (e.g., `obj` in the Stanford parser) and the *topic* is a modifier of the noun. Under this heuristic rule, a modifier can be a noun compound modifier (e.g., `nn` in the Stanford parser).

Fig. 4. Example of Heuristic 3 to Tag the Attributive Relation



For example, the sentence *"You can buy two Subway sandwiches for $7.99 on sunday."* contains an attributive relation where the noun "*sandwiches*" is the object of the verb "*buy*" and the *topic* "*Subway*" is a modifier of the noun '*sandwiches*" (a partial dependency tree is shown in Figure 4). These relations account for 16% of the development set.

Given a sentence and a topic, our rule-based classifier tries to determine if any of the 3 heuristics shown above are applicable. If this is the case, it tags the sentence as attributive.

The next section will discuss how attributive relations can be used in blog summarization and how our approach has been evaluated in that context.

4 Evaluation

To evaluate our attributive tagger, we have performed both an intrinsic and an extrinsic evaluation.

4.1 Intrinsic Evaluation

For the intrinsic evaluation, we have evaluated the performance of our attributive classifier against a manually created gold standard using precision (P), recall (R), and F-Measure (F). For this evaluation, since no standard dataset was available, we have developed our own test set containing 400 sentences from the BLOG06 corpus; where two annotators manually tagged 200 sentences as attributive and 200 as non-attributive. Discrepancy between annotators was settled through discussion to arrive at a consensus. It must be noted that both the development and the test sets contain no common sentences.

In this evaluation, we have also calculated and compared the baseline and human performance with our classifier's performance. These were computed as follows: the baseline method tags a sentence as attributive if the topic of the sentence is the direct nominal subject (i.e. heuristic rule 1 in Section 3). This method was chosen because it was the most frequently encountered dependency relation in our attributive development set (42% of the times). On the other hand, to evaluate the human performance to tag attributive relations, we asked two human participants to annotate 100 sentences from the test corpus. These 100 sentences were randomly selected from the corpus where 50 sentences are positive examples (e.g. attributive) and 50 sentences are negative examples (e.g. non-attributive). At the end, human performance was compared with the gold standard using precision, recall and F-measure.

Table 2. Intrinsic Evaluation of the Attributive Tagger

	Precision	**Recall**	**F-Measure**
Attributive Classifier	77%	76%	77%
Baseline	39%	67%	49%
Human Performance	79%	88%	83%

Table 2 shows the evaluation results of our attributive classifier. The table also shows the baseline and human performance for identifying attributive relations. We can see that the performance of the human participants (F-Measure = 83%) is much higher than the baseline (F-Measure = 49%). Our attributive classifier (F-Measure = 77%) performs better than the baseline and is a little weaker than human participants.

From the evaluation results, we can see that the precision and the overall F-Measure score of human participants are not very high (around 80%). We suspect that the reason behind this is that even though attributive relations are useful in natural language research, this relation is not well recognized and humans may not be very familiar with it. To verify this, we have calculated the inter-annotator agreement in tagging attributive sentences using Cohen's kappa. The results show that inter-annotator agreement is moderate according to [9] with a kappa value of 0.51, which seems to support our hypothesis.

4.2 Extrinsic Evaluation

To do the extrinsic evaluation, we have tested our attributive relation identification approach with our BlogSum summarizer [13] and have evaluated its effect on the summaries generated. Let us first describe the summarizer we used and how the tagger was used.

BlogSum BlogSum is a domain-independent query-based blog summarization system that uses intra-sentential discourse relations within the framework of schemata. The heart of BlogSum is based on discourse relations and text schemata.

Text schemata are patterns of discourse organization used to achieve different communicative goals. Text schemata were first introduced by McKeown [12] based on the observation that specific types of schemata are more effective to achieve a particular communicative goal. Schema-based approaches were also used by other researchers in the context of question answering and text generation to generate relevant and coherent text. However, schema-based approaches are usually domain-dependent where the domain knowledge is pre-compiled and explicitly represented in knowledge bases or is used for structured documents (e.g. Wikipedia articles).

BlogSum works in the following way: First candidate sentences are ranked using the topic and question similarity to give priority to topic and question relevant sentences. Since BlogSum works on blogs, which are opinionated in nature, to rank a sentence, the sentence polarity (e.g. positive, negative or neutral) is calculated using a subjectivity score. The subjectivity score of a sentence is also used to calculate its relevance to the question. To extract and rank sentences, our approach calculates a score for each sentence using the features shown below:

$$Sentence\ Score = Question\ Similarity + Topic\ Similarity + |SubjectivityScore|$$

where, question similarity and topic similarity are calculated using cosine similarity based on words *tf.idf* and subjectivity score is calculated using a dictionary-based approach using the MPQA lexicon[6], which contains more than 8000 entries of polarity words.

Then sentences are categorized based on the discourse relations that they convey. This step is critical because the automatic identification of discourse relations renders BlogSum independent of the domain. This step also plays a key role in content selection and summary coherence as schemata are designed using these relations. For predicate identification, BlogSum considers 28 discourse relations including the attributive relation. Then four different approaches are used to identify these predicates: a) the SPADE parser [14] (see Section 2); b) a comparison relations classifier adapted from [8] (see Section 2); c) a topic-opinion discourse relation tagger, and d) our own attributive tagger described in Section 3. It is to be noted that an analysis of 221 random summary sentences from the

[6] MPQA: http://www.cs.pitt.edu/mpqa

BLOG06 corpus shows that 32% of the sentences were tagged by our attributive tagger.

In order not to answer all questions the same way, BlogSum uses different schemata to generate a summary that answers specific types of questions. Each schema is designed based on giving priority to its associated question type and subjective sentences as summaries for opinionated texts are generated. Each schema specifies the types of predicates and the order in which they should appear in the output summary for a particular question type.

Fig. 5. A Sample Discourse Schema used in BlogSum

Predicates & Constraints
Predicate: {*Topic-opinion/Attribution*}$^{+}$
Constraint: Sentence Polarity.

Predicate: {*Contingency/Comparison*}*
Constraint: Compared Objects, Sentence Focus.

Predicate: *Attributive**
Constraint: Sentence Focus.

Figure 5 shows a sample schema that is used to answer *reason* questions (e.g. "Why do people like Picasa?"). According to this schema, one or more topic-opinion or attribution predicates followed by zero or many contingency or comparison predicates followed by zero or many attributive predicates can be used[7].

Finally the most appropriate schema is selected based on a given question type; and candidate sentences fill particular slots in the selected schema based on which discourse relations they contain.

Extrinsic Evaluation within BlogSum To evaluate the performance of our tagger in an extrinsic evaluation, we used it within BlogSum. In these experiments, we used the original ranked list of candidate sentences before applying the discourse schema, called OList, as a baseline, and compared them to the BlogSum-generated summaries with and without the tagger. We used the Text Analysis Conference (TAC) 2008 opinion summarization dataset[8] which is a subset of BLOG06. The TAC 2008 opinion summarization dataset consists of 50 questions on 28 topics; on each topic one or two questions were asked and 9 to 39 relevant documents were given. For each question, one summary was generated by OList and two by BlogSum and the maximum summary length was restricted to 250 words.

[7] Following [12]'s notations, the symbol / indicates an alternative, * indicates that the item may appear 0 to n times, + indicates that the item may appear 1 to n times.

[8] http://www.nist.gov/tac/

With this dataset, we have automatically evaluated how BlogSum performs using the standard ROUGE-2 and ROUGE-SU4 measures. For this experiment, on each question, two summaries were generated by BlogSum; one using the attributive tagger and the other without using the attributive tagger. In this experiment, ROUGE scores are also calculated for all 36 submissions in the TAC 2008 opinion summarization track. Table 3 shows the evaluation results.

Table 3. Extrinsic Evaluation of the Attributive Tagger

System Name	**ROUGE-2 (F)**	**ROUGE-SU4 (F)**
TAC Average	0.069	0.086
OList - Baseline	0.102	0.107
BlogSum without Attributive Tagger	0.113	0.115
BlogSum with Attributive Tagger	0.125	0.128
TAC Best	0.130	0.139

The table shows that BlogSum performs better than OList, and performs better with the use of the attributive tagger using both ROUGE-2 and ROUGE-SU4 metrics. Without using the attributive tagger, BlogSum misses many question relevant sentences whereas the inclusion of the attributive tagger helps to incorporate those relevant sentences into the final summary. This result indicates that our attributive tagger helps to include question relevant sentences without including noisy sentences thus improving the summary content. These results also confirms the correctness and usefulness of our tagger.

Compared to the other systems that participated to the TAC 2008 opinion summarization track, BlogSum performed very competitively; its F-Measure score difference from the TAC best system is very small. Both BlogSum and OList performed better than the TAC average systems.

5 Conclusion and Future Work

In this paper, we have presented a domain and genre-independent approach to identify attributive discourse relations which provides attributes or features of an object or an event. We have utilized dependency relations of words to identify these relations automatically. Evaluation results show that our approach achieves an F-Measure of 77% on our test-set of blogs, which compares favorably with humans and is much higher than the baseline. We have also showed that attributive relations can be used successfully in an application such as blog summarization to generate informative and question-relevant summaries.

As future work, we would like to evaluate the accuracy of each heuristic and analyze further the performance of our classifier with the goal of improving its performance and deal with attributive relations than span across clauses.

Acknowledgement

The authors would like to thank the anonymous referees for their valuable comments on a previous version of the paper. This work was financially supported by NSERC.

References

[1] Bosma, W.: Query-Based Summarization using Rhetorical Structure Theory. *In Proceedings of the 15th Meeting of Computational Linguistics in the Netherlands CLIN*, (2004), Leiden, Netherlands.

[2] Blair-Goldensohn, S.J., McKeown, K.: Integrating Rhetorical-Semantic Relation Models for Query-Focused Summarization. *In Proceedings of the Document Understanding Conference (DUC) Workshop at NAACL-HLT 2006*, (2006), New York, USA.

[3] de Marneffe, M.C., Manning, C.D.: The Stanford Typed Dependencies Representation. *In Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1–8. (2008), Manchester. U.K.

[4] da Cunha, I., SanJuan, E., Torres-Moreno, J-M., Lloberes, M., Castellón, I.: DiSeg 1.0: The First System for Spanish Discourse Segmentation. *J. Expert Systems with Applications*, 39(2):1671–1678 , 2012.

[5] Fei, Z., Huang, X., Wu, L.: Mining the Relation between Sentiment Expression and Target Using Dependency of Words. *PACLIC20: Coling 2008: Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, 257–264 (2008), Wuhan, China.

[6] Feng, V. W., Hirst, G.: Text-level Discourse Parsing with Rich Linguistic Features. *In Proceedings of the The 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-2012)*, (2012), Jeju, Korea.

[7] Grimes, J.E.: The Thread of Discourse. Cornell University, NSF-TR-1, NSF-GS-3180, 1972, Ithaca, New York.

[8] Jindal, N., Liu, B.: Identifying Comparative Sentences in Text Documents. *SIGIR'06: In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 244-251 (2006), Washington, USA.

[9] Landis, R.J., Koch. G.G.: A one-way components of variance model for categorical data. *J. Biometrics*, 33(1):671–679, 1977.

[10] Marcu, D.: From Discourse Structures to Text Summaries. *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization.* 1997, 82–88, Madrid, Spain.

[11] Marcu, D.: The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. *J. Computational Linguistics*, 26(3):395–448, 2000.

[12] McKeown, K.R.: Discourse Strategies for Generating Natural-Language Text. *J. Artificial Intelligence*, 27(1):1–41, 1985.

[13] Mithun, S.: Exploiting Rhetorical Relations in Blog Summarization. *PhD thesis*, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 2012.

[14] Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. *NAACL'03: In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 149–156 (2003), Edmonton, Canada.

[15] Taboada, M.: Discourse Markers as Signals (or not) of Rhetorical Relations. *J.Pragmatics*, 38(4):567–592, 2006.

[16] Tofiloski, M., Brooke, J., Taboada, M.: A Syntactic and Lexical-Based Discourse Segmenter. *In Proceedings of Proceedings of the 47th Annual Meeting of ACL* 2009, PA, USA.