Using the Web as Aid in Natural Language Processing

Petr Musilek

Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB T6G2V4 Canada
musilek@ece.ualberta.ca

Abstract. Association of words or phrases with their antecedents is an active field of Natural Language Processing research. It has an important role in many applications where a non-trivial level of understanding of natural language texts is desired, most notably knowledge extraction and machine translation. In a considerable minority of cases, certain pronouns can be used without referring to any specific antecedent. This phenomenon of pleonastic pronoun usage poses serious problems for systems aiming at even a shallow understanding of natural language texts. We propose a novel approach to identify such uses using a series of queries against the Web and a set of syntactic rules. The system is evaluated using several news articles with results comparable to those obtained from human efforts.

1 Introduction

Anaphora resolution [1] is an active field of Natural Language Processing (NLP) research. It plays a key role in many applications where a non-trivial level of 'understanding' of natural language texts is desired, most notably knowledge extraction and machine translation. A pronoun is a basic form of anaphor that merely refers to a previously mentioned entity or action. The ubiquitousness of pronouns, along with many other handy features, makes them relatively popular targets for researchers. However, the often underestimated problem that pronouns, especially the pronoun *it*, can be used without referring to any antecedent, actually poses a serious threat to anaphora resolution systems. The non-referential pronouns, often termed expletive or *pleonastic* pronouns, account for a non-negligible portion of all pronoun occurrences. In our analysis of the Wall Street Journal (WSJ) corpus [2], pleonastic *it* accounts for as many as 5% of all pronoun usages.

In this contribution, we focus on one kind of pleonastic *it* – the extrapositional *it*, as in '*It's a shame their meeting never took place*'. Of the three kinds of pleonastic *it* constructs (*it*-extraposition, *it*-cleft, and weather-*it*), extrapositions are by far the most widely used. Roughly 85% of all pleonastic *it* cases in our dataset are extrapositional.

© Václav Snášel (Ed.): Znalosti 2008, pp. 13–21, ISBN 978-80-227-2827-0.
FIIT STU Bratislava, Ústav informatiky a softvérového inžinierstva, 2008.

2 Related Work

The phenomenon of pleonastic pronouns has long been recognized but relatively few attempts have been made to address the problem. The existing approaches fall into two categories, rule-based [3], [4], [5], [6] and machine-learning based [7], [8], [9].

Paice and Husk [3] designed the earliest and most comprehensive rule-based system that makes use of predefined syntactic patterns and word lists. Their approach employs bracketing patterns such as '`it...that`' and '`it...who`' to meet the syntactic restrictions of extraposition and cleft. The matched portions are then subjected to further rules represented by word lists and general restrictions such as construct length and intervening punctuation. For example, construct '`it...to`' will only accept words in the task status words list, such as *easy* and *wise*. Rule-based systems rely on patterns to represent syntactic constraints and word lists to represent semantic constraints. This makes them relatively easy to implement and maintain, and fast in execution. However, it is the same features that make them less scalable - when challenged with large and unfamiliar corpora their accuracies are bound to decline.

In comparison, machine-learning based approaches, such as the one proposed by Evans [7], do not depend on word lists but rather on manually annotated training sets. Evans employs a memory-based learning algorithm with 35 features encoding positional/proximity information, part of speech, and lemmas, of both the pronoun and other sentence components of interest. Machine-learning based approaches are partly able to get around the restrictions imposed by fixed word lists. However, learning also comes with a price – retraining may be necessary for different domains. Moreover, features used for learning are still unable to realistically and reliably capture the semantics of the original sentences, hindering the results of feature-based approaches.

3 A Web Based Approach

In determining whether an *it* instance is extrapositional, both syntactic patterns and semantics of various clause constituents play important roles. The goal of the system is to obtain high precision and maintain good coverage at the same time. To achieve this, it attempts to make good use of both syntactic and semantic information available in the text. A set of relaxed, yet highly relevant, syntactic patterns is first applied to the input text to filter out syntactically inviable cases (cf. [10] for discussions of the syntactic signatures of *it*-extrapositions). Unlike the matching routines of some previous approaches, with a few exceptions of heuristics, this process avoids detailed prescriptions of syntactic patterns and tries to include every piece of text that poses a possible extraposition. The candidates are then subjected to different semantic tests performed as queries against the Web. Results of the queries provide a direct evidence of how the specific configuration of constructs is generally used. The process is illustrated in Fig. 1.

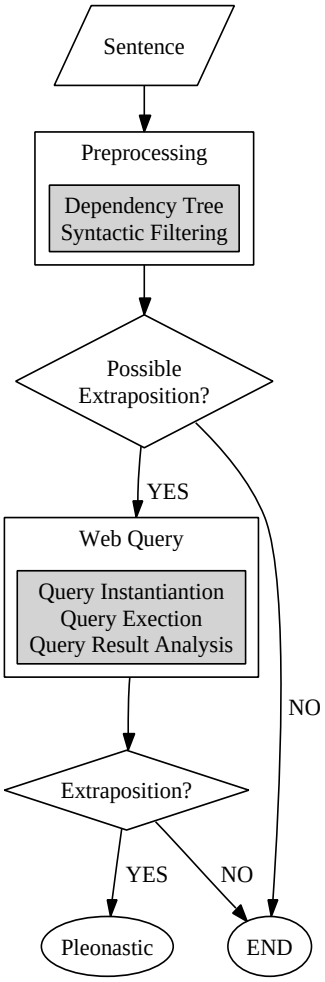

Fig. 1. The process of Web based identification of extrapositional *it*

The reasons that such a corpus-based approach is chosen, as opposed to manually constructed knowledge sources such as the WordNet [6] or a word list, are threefold:

- first, manually constructed knowledge sources, no matter how big they are, contain only a small portion of general world knowledge;
- second, the manually constructed knowledge sources are subject to specific ways of organization, which may not satisfy the system's needs;
- finally, human languages are flexible and subject to changes that are naturally reflected in corpora but do not easily propagate to manually constructed knowledge bases.

As a corpus, the Web's virtually unlimited size provides some additional benefits. As the largest collection of texts in natural language, it not only hosts a considerable portion of general world knowledge, but also stores this information using the very syntaxes that define our language. Because of its broad coverage, the overall likelihood of the Web covering an individual's knowledge, both general and language-usage specific, is much higher than that of a compiled corpus. In addition, it is devoid of the systematic noise introduced into manually constructed knowledge sources during the compilation process (e.g. failure to include less frequent items or inflexible ways of information organization). Overall, the Web is a statistically more reliable instrument for analyzing various semantic relationships stored in natural languages by means of examples.

3.1 Design of Search Engine Queries

The system employs two sets of query patterns, the *what*-cleft and the comparative analysis, each providing a unique perspective of the sentence in question.

The *what*-cleft pattern,

```
What + verb phrase + copula + stub                (1)
```

is a *what*-(pseudo-)cleft construct that encompasses matrix-level information found in an *it*-extraposition. The pattern is obtained by first transforming the original sentence into its non-extraposed format, which is then further transformed into a *what*-cleft. For example:

```
It is easy to see why the ancient art is on the ropes. ->
Why the ancient art is on the ropes is easy to see.    ->
What is easy is to see why the ancient art...
```

Since *what*-cleft is not a very popular construct, it is necessary to further simplify the query in order to expand its coverage. This is achieved by reducing the subordinate clause to a stub. After simplification the query now becomes

```
What is easy is to
```

The choice of stub depends on the structure of the original subordinate clause: *to* is used when the original subordinate clause is an infinitive, a gerund, or a *for...* infinitive construct. For all other cases, the original subordinate conjunction, or *that* in case there is no conjunction, is used as stub. The use of a stub in the pattern imposes a syntactic constraint, in addition to the ones prescribed by the pronoun *what* and the copula *is*, that demands a subordinate clause be present in query results. The choice of stubs also reflects to a certain degree the semantics of the original texts and therefore can be seen as a weak semantic constraint.

The second pattern, the comparative expletiveness test, provides a 'simplified' account of the original text in a few different flavors. The general form of the pattern is as follows:

```
pronoun + verb phrase + simplified extraposed clause      (2)
```

For example,

```
It is easy to see why the ancient art is on the ropes. ->
{ it                    }
{ which/who/this/he     }  is easy to see why the
```

The only difference among the individual patterns is the choice of pronoun as the matrix clause subject (i.e. *it*, *which*, *who*, *this*, and *he*). Because of the expletive nature of the pronoun in an *it*-extraposition, replacing it with other pronouns will render the sentence invalid. Therefore when the pattern is instantiated and submitted to a search engine, the number of hits obtained from *it* version should by far outnumber that of the other versions combined if the original text is an it-extraposition; otherwise the number of hits should be comparable.

Similar to the case of the *what*-cleft pattern, how each extraposed clause is simplified also depends on its original structure, as listed in Table 2.

Original Structure	**Simplified**
infinitive	infinitive + stub
for... infinitive	infinitive + stub
gerund	gerund + stub
full clause led by subordinate conjunction	conjunction + stub
full clause without subordinate conjunction	*that* + stub

Table 1. Simplified Extraposed Clause

The stub can be either *the*, which is the most widely used determiner, or a combination of various determiners, personal pronouns and possessive pronouns, all of which indicate a subsequent noun phrase. In case an infinitive construct involves a subordinate clause led by a *wh*-adverb or *that*, the conjunction is used as stub. This arrangement guarantees that the results returned from the query conform to the original text both syntactically and semantically. A null value should be used for stubs in an object position if the original text lacks a nominal object.

3.2 Improving the System's Coverage

Both patterns introduced in Section 3.1 are highly specific to extrapositions. In other words, they seldom classify a case that is in reality not an extraposition as extrapositional. However, even with the various simplifications that are already made, neither pattern is sensitive enough. For many valid extrapositional cases the queries fail to yield any meaningful results.

The lack of sensitivity is hardly surprising – in human languages, there are virtually numerous different ways to enhance or qualify any concept. For example, instead of simply saying '*it is easy (to implement a system like this)*', one may use '*it is quite easy*'. Similarly, '*it really helps*' is a perfectly valid alternative for '*it helps (to have user feedback)*'. Such differences in expression usually

have no effect on the judgement of the pronoun's expletiveness. However if the additional components are included in the query, it could significantly reduce the scope of the search and often cause the query to return no result at all.

If all sentences on the Web were already parsed, it would have been trivial to get around the problem because the irrelevant items in both the query and the target can be conveniently ignored. Unfortunately, at least for now, the only way to query the Web is through primitive pattern matching. The best approach under this restriction is then to transform the original expression into one that is relatively more popular, and use the new expression in the query. The transformation process provides different renditions based on the structure of the original object:

- Adjective phrases:
  Head word only, with the exception of the adverbs *too* and *not*, which are retained in order to better entertain the *too... to* construct and to maintain compatibility with the semantics of the original text.
- Common noun phrases:
  - with a possessive ending/pronoun, or an *of*-preposition:
    `$PRPS$` plus the head word, where `$PRPS$` is a list of possessive pronouns. For example, '*his location*' can be expanded to '*its* | *my* | *our* | *his* | *her* | *their* | *your location*'.
  - with determiners:
    `$DT$` plus the head word, where `$DT$` is a list of determiners (e.g. *a*, *any*, *the* etc.) and/or demonstratives (e.g. *this*, *those* etc.) chosen based on the configuration of the original text.
  - without determiner:
    Head word only.
- Proper nouns and pronouns:
  `$PRP$`, which is a list of personal pronouns.
- Prepositional phrases:
  Preposition plus `$PRPS$` plus `truncated prepositional object`, where the original object of the preposition is truncated in a recursive operation.
- Numeric values:
  '*a lot*'

Verbs are also expanded to include both the simple past tense and the third person singular present form with the aid of WordNet and some generic patterns. Where applicable, particles such as *out* and *up* remain attached to the verb after the transformation. Taking the sentence '*It's a shame their meeting never took place.*' as an example, after the transformation the three queries instantiated from the *what*-cleft pattern and the comparative expletive pattern are as follows:

```
what  is|was|'s  a|an|no|any  shame  is|was  that
it                is|was|'s  a|an|no|any  shame that  the|a|this
which|this|who|he  is|was|'s  a|an|no|any  shame that  the|a|this
```

Aside from transforming the original texts, a stepped-down version of the comparative expletiveness pattern is also provided to further enhance the system's coverage. The current scheme is to simply replace the `simplified extraposed clause` with a new stub – *to* – if the original extraposed clause is an infinitive, a *for*... infinitive, or a gerund construct. For example,

`It is easy to see why the ancient art is on the ropes. ->`
$\left\{ \begin{array}{l} it \\ which/who/this/he \end{array} \right\}$ `is easy to`

In other situations, no downgraded version is provided.

4 Binary Classification of *It*-extraposition

Five queries (one instantiated from the *what*-cleft pattern, two from the comparative expletive test pattern, and two from the stepped-down comparative expletive test) are executed for each sentence that has syntactic features similar to those of *it*-extrapositions. The number of results returned from the queries are denoted n_w, n_{it}, n_{others}, n'_{it}, and n'_{others}, respectively.

Almost all *it*-extrapositions have a valid *what*-cleft reading, and vice versa. Therefore, if the *what*-cleft query yields sufficient number of results (i.e. when n_w is greater than a threshold, N_w) it is likely that the original sentence is extrapositional. In theory, n_w should be a very specific indicator for *it*-extrapositions, which hints that the threshold should be set close to 0. However during preliminary experiments it was found that common typos often cause the query to return a small amount of results. Consequently the threshold is set to $N_w = 100$ instead.

Two new variables, $r = (n_{others} + \beta)/(n_{it} + \beta)$ and $r' = (n'_{others} + \beta')/(n'_{it} + \beta')$, are defined to serve as indicators of wether the pronoun *it* can be replaced by another pronoun or demonstrative. Since none of the alternatives can replace an extrapositional *it*, the ratios should be close to 0 if the original sentence is an extraposition. Based on results of preliminary experiments, $R_{ext} = 0.15$ is chosen as the threshold for r and r' to indicate likely extraposition. In addition, $R_{ref} = 1$ is also set up as the absolute upper limit beyond which the *it* instance is much more likely referential. The parameters $\beta = 1$ and $\beta' = 10$ serve to disqualify cases that produced too few results for a meaningful comparison. A larger value is assigned to β' because the stepped-down queries usually yield more matches.

The final classification E is defined as follows:

$$E = \begin{cases} (r < R_{ext} \text{ AND } r' < R_{ref}) & \text{if } max(n_{others}, n_{it}) \geq N_{min}, \\ (n_w > N_w \text{ OR } r' < R_{ext}) & \text{if } max(n_{others}, n_{it}) < N_{min}, \end{cases} \quad (3)$$

where $N_{min} = 10$ indicates the minimum required number of results.

5 Evaluation

For the purpose of this study, the first 1000 occurrences of *it* from the WSJ corpus are selected as the development dataset, upon which the system's syntax processor is developed. The values of various constants are also determined using this dataset. An additional 500 instances are also randomly chosen from the rest of the corpus to serve as the test dataset. Both sets have been manually annotated.

Dataset	Instances	Identified	Correct	Precision	Recall	F-measure
Development	118	114	112	98.25%	94.92%	96.55%
Test	63	59	58	98.31%	92.06%	95.08%

Table 2. System Performance

In comparison, a re-implementation of the Paice and Husk algorithm only correctly recognizes 90 out of the 118 cases in the development dataset (76.27% recall). It is difficult to assess the algorithm's precision because some of the used patterns are also applicable to other forms of pleonastic *it*; however it is likely around 54% (which is the algorithm's overall precision across multiple pleonastic subtypes). On the test dataset, the Paice and Husk algorithm yields similar performances (76.19% recall for extrapositions and 57% overall precision).

Many of the extrapositional cases are also annotated in the WSJ corpus. Unfortunately, in each dataset there is at least one item that is indisputably misclassified. Overall, the proposed approach is able to operate at the same level of precision as that of the WSJ human-made annotations.

6 Conclusion and Future Work

In this contribution a new approach to identify *it*-extrapositions has been introduced. Unlike previous approaches covering this phenomenon, the new algorithm exploits the Web to find out whether an instance of *it* is expletive. The original sentences are restructured and multiple queries are generated according to two patterns. After the queries are executed using a search engine, the result counts are analyzed to make the final conclusion. The concept of the system is very simple, and it does not require either manually constructed word lists or manually annotated training data. Evaluations also suggest that the system is highly accurate, with precision comparable to human judgement. Given this success, it would be very interesting to find out how the same concept can be extended to other areas of natural language processing, for example anaphora resolution and parsing (syntax tree generation).

Acknowledgements

Support from the Natural Sciences and Engineering Research Council (NSERC), Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Appplications" and by Information society project No. IET100300414 are gratefully acknowledged.

References

1. Mitkov, R.: Outstanding issues in anaphora resolution. In Gelbukh, A., ed.: Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2001). Volume 2004 of Lecture Notes in Computer Science., Berlin, Springer (February 2001) 110–125
2. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics **19**(2) (June 1993) 313–330
3. Paice, C.D., Husk, G.D.: Towards the automatic recognition of anaphoric features in english text: the impersonal pronoun "it". Computer Speech & Language **2**(2) (June 1987) 109–132
4. Lappin, S., Leass, H.J.: An algorithm for pronominal anaphora resolution. Computational Linguistics **20**(4) (December 1994) 535–561
5. Denber, M.: Automatic resolution of anaphora in English. Technical report, Eastman Kodak Co. (1998)
6. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, Mass., USA (1998)
7. Evans, R.: Applying machine learning toward an automatic classification of *it*. Literary and Linguistic Computing **16**(1) (April 2001) 45–57
8. Boyd, A., Gegg-Harrison, W., Byron, D.: Identifying non-referential *it*: A machine learning approach incorporating linguistically motivated patterns. In: Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing, Association for Computational Linguistics (June 2005) 40–47
9. Mitkov, R., Evans, R., Orasan, C.: A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In Gelbukh, A.F., ed.: Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2002). Volume 2276 of Lecture Notes in Computer Science., London, UK, Springer-Verlag (2002) 168–186
10. Kaltenböck, G.: *It*-extraposition in English: A functional view. International Journal of Corpus Linguistics **10**(2) (2005) 119–159