

Weight Derivation for Saliency Algorithms in Pronominal Anaphora Resolution (working paper)

Aleem Hossain and Mark Lee

School Of Computer Science

University Of Birmingham

Edgbaston

Birmingham

B15 2TT

[ug13axh, mgl]@cs.bham.ac.uk

Abstract

Work is presented which describes research carried out over a ten-week period. Whilst still ongoing, the paper here describes the steps taken towards creating an effective pronoun resolution system, based on that of Lappin and Leass. Existing implementation, difficulties and proposed future additions to this project are described in full. This paper also sets out to describe briefly some of the theory behind pronominal anaphora resolution and previous work in the field.

Introduction

An important problem in natural language processing is the resolution of pronouns to their intended referents.

Consider the sentence:

- (1) John gave Tom the ball and when John nodded his head, Tom kicked it.

This sentence demonstrates an anaphor, where the pronoun ‘it’ refers back to a referent. Intuitively, ‘it’ refers to ‘the ball’. The task for a natural language system is to work out that ‘it’ does indeed refer back to ‘the ball’ and not to ‘[John’s] head’. (It should be noted that ‘his’ is also anaphoric, but it is “it” which shall be of interest here).

One such algorithm for the resolution of pronouns was proposed by Lappin and Leass (1994). This algorithm is based on the fact that pronouns are more likely to refer to entities mentioned recently in the discourse (usually no further than one or two sentences back). The algorithm involves calculating saliency values for each new entity that is encountered in a noun phrase. These saliency values are calculated by

summing the weights assigned to various factors. Such factors might include sentence recency, indirect object and oblique complement emphasis, head noun emphasis (referents embedded in long noun phrases) and non-adverbial emphasis. Higher weights are assigned to factors that are deemed to be more “useful” when deciding how to resolve a pronoun.

Once saliency values have been calculated for each referent, the algorithm can be applied to resolve the pronouns. The entity with the highest saliency value is declared to be the most likely referent. Where two or more entities are awarded the same saliency value, the most recent one is used. The weights used in the saliency algorithm are ad hoc. This research was concerned with implementing a pronominal resolution system and adjusting these weights empirically using machine-learning techniques based on real corpus data.

The remainder of this paper will give the theoretical details of anaphora; provide an overview of Lappin and Leass’ (ibid) anaphora resolution algorithm; discuss other work carried out with refined saliency weighting; describe the work already carried out and future development by this author and finally, briefly look at similar research.

Anaphora

Consider again example (1). The entity referred back to (in this case ‘the ball’) is called the ‘referent’ or ‘antecedent’. ‘It’ is called the ‘referring expression’ or ‘anaphor’; that is, the expression used to perform reference. Reference to an entity that has been previously introduced into this discourse is called anaphora. Determining the antecedent of an anaphor is called anaphora resolution.

When a referent is initially mentioned, a representation for it is evoked. Subsequent

mentions can be accessed from the discourse model. Cohesion is not a structural relation, so is unrestricted by sentence boundaries. However, pronouns refer to entities evoked no further than two sentences back (whereas definite noun phrases, for example, can refer further back). Sentences where the antecedent is in the same sentence as the anaphor are called intrasentential, whereas intersentential refers to antecedents that are in a different sentence to the anaphor.

When performing anaphora resolution, all noun phrases are typically treated as potential candidates for antecedents. The scope is usually limited to the current and preceding sentences and all candidate antecedents within that scope are considered.

In the English language, all antecedents and anaphor agree in gender and number. It is eliminating factors (or constraints) such as these that can be used in algorithms such as Lappin and Leass' (ibid). Other constraints include person and case agreement; syntactic constraints, for instance "Mark bought him a present" [him≠Mark] and selectional restrictions. For example:

(2) John parked his Acura in the garage.
He had driven it around for hours.¹

Whilst "it" could be "his Acura" or "the garage", the verb "drive" requires an object that can be driven. However, selectional constraints are complicated by metaphor such as:

(3) John bought a new Acura. It drinks
petrol like you would not believe.¹

Obviously, cars cannot literally drink petrol, but the metaphor allows "it" to refer to the "new Acura".

The Algorithm

This section attempts to summarise Lappin and Leass' algorithm (ibid). The algorithm exploits constraints such as those mentioned in the previous sentence, and also uses rules about other preferences that can be used when trying to perform pronominal resolution. These

salience factors and their associated weights are sentence recency (100); subject emphasis (80); existential emphasis (70); accusative emphasis (50); indirect object and oblique complement emphasis (40); non-adverbial emphasis (50) and head noun emphasis (80).

When a new entity is evoked, these weights can be used to calculate its salience value. The salience value is simply the sum of the associated weights. If there are no pronouns to be resolved in a sentence, the next sentence is processed and the weights that contribute to an entity's salience are halved (to account for sentence recency).

It is possible that the same referent is mentioned in multiple sentences (each with different salience values) and so the contributions of each need to be considered. As such, if a potential referent is mentioned in the current sentence and a previous one, then the weight for each will be taken into account (though, if the referent appears more than once in the *same* sentence, then the weight will only be considered once). Consequently, the idea of repeated mention is taken care of. This theory says that entities that have been the focus previously, are more likely to be referred to by pronouns in subsequent discourse.

When a pronoun is encountered, the referents that do not abide by the constraints already indicated are removed. At this stage, two new salience weights are introduced. They are role parallelism (35) and cataphora (-175).² Finally, of the referents that remain, the one with the highest salience value is deemed to be the result.

Weight Derivation

The work initiated as outlined in this paper is being continued and extended as part of the author's final year project.

The author hopes to resolve the problems with the grammar that is being used with the chart parser quickly and then complete the weight summation and anaphora resolution algorithm.

Once an implementation of Lappin and Leass' algorithm (ibid) is functional and has

¹ Adapted from Jurafsky & Martin (2000) p681

² Both these factors require a pronoun in order to be calculated.

been tested and evaluated, work will begin to refine the salience weights. Additionally, it is hoped to train Brill's tagger on British National Corpus (BNC) data to obtain a BNC tagger.

It is anticipated that a genetic algorithm (GA) will be used to refine the salience weightings. GAs are an efficient way of exploring large search spaces. A non-rigorous, general algorithm might be:

- (1) Initialise an initial population of weights.
- (2) Evaluate their fitness
- (3) Select parents based on (3)
- (4) Produce offspring from using crossover and mutation techniques
- (5) Repeat (2) to (4) until the stopping criteria is reached

Work will have to be done to design a suitable representation for the weights. Other variables such as population size, crossover operator and mutation weights will also have to be considered when designing such an algorithm.

Each set of weights will be tested on real corpus data to evaluate their fitness. It is hoped that the derived weights will improve the performance of the pronominal anaphora resolution algorithm as implemented initially.

Refined salience weighting

The motivation of this research was to successfully implement Lappin and Leass' algorithm (ibid) and then to proceed by adjusting the weights using machine-learning techniques.

Similar work has been carried out on the algorithm used by MARS, a pronominal anaphora resolution system, and this work is presented in (Evans, 1998). Here, weights are derived using a genetic algorithm. GAs are an extremely effective machine-learning technique since they allow a very large search space to be explored. This makes them ideal for optimising parameters such as the weights being described here. Evans (ibid) describes how parameter sets (salience weights) were generated, evaluated and then adjusted. This process was repeated over 300 generations. Evans (ibid) describes a biased roulette wheel selection process whereby fitter sets were promoted for further processing; new

generations were obtained using mutation and crossover operators. Evans (ibid) found that the GA did indeed evolve weights that improved the performance of MARS.

It is this type of research that the author of this paper would like to carry out. It is believed that refined salience weighting using machine-learning techniques on Lappin and Leass' algorithm (ibid) is a unique approach that will alter the weights to be calculated empirically.

Implementation

The system has been developed in Java under a UNIX environment. Currently, the program reads in a corpus in the form of a text file. The corpus is split into sentences and tokenised. This is done by scanning the input corpus sentence-by-sentence for so-called "end-of-sentence markers" such as '!' (*exclamation mark*); '.' (*full stop*) and '?' (*question mark*). Any punctuation encountered in the corpus is preceded by white space so that "Don't" becomes "don 't". Tokenisation of this kind is required to aid the tagging process, for instance, to allow for the tagging of the possessive ending "s". Furthermore, the Penn Treebank part-of-speech tagset (which is used by the tagger) includes entries for characters such as "" (*left quote*).

The tagger incorporated into the system is that implemented by Brill (1995). This is a transformation-based tagger that uses both a set of rules to determine when a particular tag should be assigned to an ambiguous word³ and also includes a machine-learning component to induce rules from a previously tagged corpus. Currently, the tagger uses the Penn Treebank tagset and has been used on the Brown corpus. It is hoped, though, that future work will allow the training of this tagger on British National Corpus (BNC) data to produce a BNC tagger.

After tagging, discourse entities are extracted from the resulting tagged corpus. This is achieved using a series of regular expressions. Entities of the form "determiner...noun" "pronoun...noun" and "standalone nouns" are created. Each entity is treated separately. It is

³ a word which could have more than one tag assigned to it

this point of the implementation that presented the most difficulties.

In order to aid salience calculation, an Active Chart Parser was implemented. A grammar capable of parsing entries from the Penn Treebank was provided by Paul Llado of Stanford University. Unfortunately, there have been difficulties in incorporating this grammar successfully. It is hoped, however, that this issue will be resolved soon and so details of this grammar will be documented in this author's final paper.

Known Limitations

The process for splitting the input corpus into sentences has been described. However, the system is not currently capable of handling abbreviations which are often followed by so-called "end-of-sentence" markers. For example, the single sentence "Dr. Jones issued me with a prescription." would be separated in this way:

Dr.
Jones issued me with a prescription.

Obviously, this should be treated as a single sentence.

Evaluation

Due to time constraints and implementation difficulties, a working pronominal resolution system has not been achieved. Failure to plan the implementation of the software properly caused significant delays in its development. Additionally, difficulties in effectively and efficiently extracting discourse entities from the corpus and then proceeding to calculate salience scores to each entity meant that the work could not be completed within the initial 10-week scope (which was the intended length of this brief project).

Future extensions could include adapting and refining the system to deal with spoken dialogue. Work to resolve pronouns in this way is described by Strube and Müller (2003). This would involve allowing the system to take in a voice input (or a corpus of spoken dialogue) and being able to deal with the poor performance of human speech (for example, repeated words and hesitations such as "erm").

In order to progress this project effectively, more time must be spent on the design of the remainder of the system.

Related Work

Modified versions of Lappin and Leass' algorithm (ibid) do exist. Kennedy & Boguraev (1996) describe a variant that does not require parsing of text. Lappin and Leass' algorithm (ibid) requires parsing in order to determine the role that each entity plays in any given sentence. The work presented in (Kennedy & Boguraev, 1996) used the output of a part-of-speech tagger with added information about grammatical function. They show their results to be comparable to those reported in (Lappin & Leass, 1994).

Conclusion

The author has outlined the beginnings of a substantial research project. Background information and an initial literature review was undertaken. Details of the implementation in its current existence were described. Moreover, the current limitations of the implementation were examined. Difficulties encountered so far were outlined and observations drawn to ensue future progress. The author then outlined future work to be carried out on this project.

Acknowledgements

The work documented in this paper was carried out and funded thanks to the award of a 10-week summer research scholarship by The School of Computer Science, University of Birmingham.

Thanks also to Paul Llado of Stanford University for providing the grammar which, it is hoped, will be incorporated into the chart parser.

References

- Aston, G. & Burnard, L. (1998) *The BNC Handbook. Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press
- Biber, D., Johansson, S., Leech, G., Conrad, S. & Finegan, E. (1999) *Grammar Of Spoken And Written English*. Essex: Longman

Evans, R. (2002) "Refined Salience Weighting And Error Analysis In Anaphora Resolution." *Proceedings of the Workshop on Reference Resolution and Natural Language Processing (RRNLP'02)* Alicante, Spain

Halliday, M.A.K. & Hasan, R. (1976) *Cohesion In English*. Essex: Longman

Jurafsky, D. & Martin, J.H. (2000) *Speech And Language Processing*. New Jersey, USA: Prentice-Hall.

Kennedy, C. & Boguraev, B. (1996) "Anaphora For Everyone: Pronominal Anaphora Resolution Without A Parser." *Proceedings of the 16th international conference on Computational Linguistics (COLING'96)* pages 113-118, Copenhagen, Denmark

Mitkov, R. (1999) *Anaphora Resolution: The State Of The Art*. Working paper, School of Languages and European Studies, University of Wolverhampton

Mitkov, R. ed. (2003) *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press

Pinker, S. (1995) *The Language Instinct*. London: Penguin

Shalom, L. & Leass, H. (1994) "An Algorithm For Pronominal Anaphora Resolution" *Computational Linguistics*, **20**(4):525-561

Strube, M. & Müller, C. (2003) "A Machine Learning Approach To Pronoun Resolution In Spoken Dialogue." *Proceedings of 41st annual meeting of the Association for Computational Linguistics (ACL'03)* Sapporo, Japan

Trask, R.L. (1999) *A Dictionary Of Grammatical Terms In Linguistics*. London: Routledge