Pronoun Generation for Text Summarization and Question Answering

Mehdi M. Kashani, Fred Popowich

School of Computing Science
Simon Fraser University
8888 University Drive, Burnaby, BC, Canada
{mmostafa, popowich}@sfu.ca

Abstract

An algorithm for pronoun generation is introduced as part of a summarization and question answering system. The algorithm makes use of Lingpipe, a coreference resolution tool, as a key component of a process to generate the appropriate pronouns. The two phased algorithm makes use of a replacement phase, followed by a validation phase which makes use of information obtained by a parser. At the end, initial results performed on a collection of DUC 2005 documents are provided.

Tvorjenje zaimkov v sistemih povzemanja besedila in odgovarjanja na vprašanja

Predstavljen je algoritem za tvorjenje zaimkov kot del sistema povzemanja in odgovarjanja na vprašanja. Algoritem uporablja LingPipe kot ključno komponento postopka za tvorjenje ustreznega zaimka. Dvostopenjski algoritem uporablja fazo zamenjave, ki ji sledi faza validacije, pri kateri se uporablja informacije, pridobljene s avtomatskim razčlenjevanjem. Na koncu so predstavljeni prvi rezultati delovanja sistema na zbirki dokumentov DUC (Document Understanding Conferences).

1. Introduction

Summarization and question answering are both examples of natural language processing systems that produce natural language output, and thus require some sort of text generation module. The degree of sophistication in the text generation can vary widely, but given the high frequency of pronouns in natural language text, it is natural to expect that a proper treatment of pronouns in summaries and responses might lead to better quality output. We examine this issue by exploring an approach to pronoun generation which incorporates a pronoun *resolution* module as part of the generation process.

Little attention has been paid to pronoun generation and the focus has always been on coreference resolution. The reason can be attributed to the lack of a good benchmark for evaluation and/or scarcity of real language generation systems that the pronoun generation module can be plugged into.

Sometimes resolution algorithms can be viewed as clues to generation. The first rule of a centering model proposed by Grosz et al (1995) can be interpreted as an acceptance criterion for pronoun generation. However, this is only a special case and no one has really implemented the idea in a generation framework.

McCoy et al. (1999) hypothesize that discourse structure is indeed vital in the decision of whether or not to generate a pronoun. To prove their claim, they choose the shift in time scale as a signal of change in the deictic center of the story. Based on time clues, they segment the text into different threads and in their algorithm state that if the current and previous references to X are in the same thread a pronoun is preferable and otherwise a definite description is used. In case of ambiguities, they use a reference resolution algorithm (Strube, 1998) and check if the pronoun would resolve to X in which case it is permitted to use a pronoun, otherwise not. By using these rules, they show a reduction of error rate by 28.9% compared to a baseline. We take a similar and simpler approach for another task and show an improvement.

This paper introduces a pronoun generation approach used in the course of a summarization and question answering system (Melli, 2005). The task was to find answers, less than 250 words, to fifty questions using the given corpus of relevant documents. These documents came from the Financial Times of London and from the Los Angeles Times.

The general approach taken in the summarization task involved a linguistic analysis of each sentence in the corpus, performing not only named entity extraction, but also anaphora resolution, in which each pronoun in the document was tagged with the entity corresponding to its antecedent. Sentences were then selected from the various documents, and combined to form a summary. Note that by replacing pronouns with their antecedents, it allows the use in the summary of a sentence that originally contained a pronoun, even if the sentence containing the antecedent is not included in the summary. Thus, “dangling” pronouns are avoided.

As is measured in (Vicedo and Ferrandez, 2000) the ratio of pronominal reference used in news collections can be as high as 55%. So, to make the final text smooth and fluent, pronoun generation is essential.

In this paper we focus only on third person singular pronouns where they are in contexts handled by Lingpipe[1]. In the task for which we used our approach, sentences are wholly extracted from original documents. Reflexive pronouns and cataphora are generally both intra-sentence concerns and their generation is not needed in the task. Also note that first and second person pronouns are never considered in pronoun generation systems, because their generation requires the change of sentence structure, something which is usually not desired (for example verbs should change as well). The cases where the proper noun acts as an adjective (as in "the

[1] Lingpipe is a suite of natural language processing tools written in Java that performs tokenization, sentence detection, named entity detection and co-reference resolution on text. The input is plain text and output is an XML file with embedded tags inside the original text.

Castro government") are not dealt with in this paper either.

In this paper, we first explain in section 2 the algorithm in detail and in section 3 our results are provided. We conclude in section 4 with some suggestions for possible enhancements.

2. Our Algorithm

As mentioned above, Lingpipe can find all the referents to a specific entity, so by running Lingpipe once on the text, we would have a chain of entities, all with the same referent. Our goal is for the latter entities to be systematically replaced by pronouns referring to the former entities. The algorithm can be divided into two phases: *replacement* and *validation*.

In the first phase, an appropriate pronoun is chosen and the text is regenerated with the specific entity replaced by this pronoun. Then, Lingpipe is used to validate the replacement. In case of valid replacement, the pronoun will remain in the final text.

Nearly all of the existing algorithms for anaphora resolution identify a part of the text surrounding the pronoun that will be inspected for the candidate antecedent. Lappin and Lease (1994) and Mitkov (1998) use the preceding three and two sentences respectively. Gaizauskas and Humphreys (1996) use the same paragraph that the pronoun is located. As sentences in news articles tend to be long, we chose the distance as at most two sentences. The sentence boundary detector in Lingpipe was used for this task.

In order to suggest a pronoun out of the pronoun set (he, she, his, her, him), we have to deal with gender and case (nominative, accusative, possessive) as part of the replacement phase. Since Lingpipe is not able to guarantee grammaticality, we cannot deal with grammaticality when we use it in the validation phase.

The gender recognition task is itself performed in four consecutive phases. The first, third and fourth phase are general and the second phase takes advantage of the information available in previous stages in the pipeline.

First, the summary is checked to see if we can resolve gender using existing referring pronouns. Second, in the annotated document set, named entity information for all the original documents exists and is used to extract gender information. Third, if some entities remain unresolved (either because they are not referenced by a pronoun or the co-reference is not detected by Lingpipe) an online database of 10079 international frequently used names[2] is used. Fourth, the prefix courtesy titles (ex. Mr) are applied, overriding all of the above. If the gender of an entity cannot be distinguished after these four phases, its gender is marked as *male* (due to the dominance of male entities in news articles).

In order to choose between different types of pronouns (nominative, accusative, possessive) the information available from the parse is used. Specifically, the following rules are applied:

1. If most of the prepositions precede the entity and it is not followed by 's, the replaced pronoun should be accusative (him, her). These prepositions do not include all of the words labelled as PP in the parser[3].
2. If most of the prepositions precede the entity and the entity is followed by 's, the replaced pronoun should be possessive (his, her).
3. If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is not followed by 's, the replaced pronoun should be accusative (him, her).
4. If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is followed by 's, the replaced pronoun should be possessive (his, her).
5. In all other cases, the replaced pronoun is nominative and based on gender information (he, she).

After the pronoun is replaced in the text, the text is fed to Lingpipe. This new output is compared with the original text. If the new pronoun is still referring to the same entity that the earlier entity referenced (i.e. the entity that is replaced by the pronoun), the replacement will be valid and the pronoun is kept in the text, otherwise the previous version of the text is used for the next replacement iteration. This process is repeated for all the possible combinations of co-referent entities. If the entity is already a pronoun, nothing is done**.**

3. An Example

To illustrate how this algorithm works, we will now work through an example. Suppose the following passages, shown in (1), (2) and (3), are extracted from three separate documents[4]:

(1) Albert lives alone.
(2) Sandra invited Albert to the dinner.
(3) Jack couldn't make it to the party. Albert is in a hurry.

By running Lingpipe on the set, it would return (Albert, Sandra and Jack) as the entities. The output would be as shown in (4-6).

(4) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(5) <ENAMEX id="1" type="PERSON"> Sandra </ENAMEX> invited <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> to the dinner.
(6) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> is in a hurry.

So there are three co-referent *Alberts*, one *Sandra* and one *Jack*. As the algorithm states, there is an opportunity for the second and third *Alberts* to be replaced by pronouns. First, the gender recognition task is performed and after the four phases explained in the section 2 the genders would be known.

[2] http://baby-names.adoption.com/names.php

[3] For instance, while nominative pronouns can occur after *while*, *while* is categorized as prepositional phrase.

[4] This example is not extracted from the DUC2005 corpus for the sake of simplicity. Also, it is not tested by the implemented code and Lingpipe. The purpose is just to show how the algorithm works.

Since there are at most two potential replacements, the loop runs twice. On the first run, *him* is suggested instead of the second *Albert* (following the 4th rule proposed in section 2). Then, the following text is generated:

(7) Albert lives alone. Sandra invited him to the dinner. Jack couldn't make it to the party. Albert is in a hurry.

Notice the only change to the text is the introduction of this single pronoun. Now, this text is fed to Lingpipe to generate the following output:

(8) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(9) <ENAMEX id="1" type="PERSON">Sandra</ENAMEX> invited <ENAMEX id="0" type="MALE_PRONOUN"> him </ENAMEX> to the dinner.
(10) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> is in a hurry.

In the validation phase, the id of the newly-replaced pronoun (*him*) is compared with the entity it was replaced with (the second *Albert*). Since both of them are 0, it means the pronoun is correctly referring to the antecedent of the replaced entity, so it is kept in the final text.

On the second pass, the third *Albert* is replaced with a pronoun as shown in (11).

(11) Albert lives alone. Sandra invited him to the dinner. Jack couldn't make it to the party. He is in a hurry.

After running Lingpipe on the text, we obtain the following:

(12) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(13) <ENAMEX id="1" type="PERSON"> Sandra </ENAMEX> invited <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> to the dinner.
(14) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="2" type="MALE_PRONOUN"> He </ENAMEX> is in a hurry.

The id of *he* is 2 not 0, meaning that Lingpipe suggests if we perform such a replacement we end up referring to *Jack* instead of *Albert* which is not author's purpose. So, this replacement is rejected.

4. Results

We use the DUC 2005 documents for our evaluation. One issue we encountered with the DUC2005 questions was that they were not really person-centric and since the final answer was heavily dependent on the question keywords, we could not have recurrent person entities in them.

We decided to use the current project corpus but to overcome the lack of test data by extracting the sentences containing the same entity and order them randomly and to occasionally insert some other sentences between them. So, the following results are based on the DUC2005 corpus but are not using the DUC2005 questions. This approach might seem artificial but is consistent with questions like “who is X?”, where the only significant keyword is the name of the person. McCoy et al (1999) showed in 97.9% cases not using the pronoun for long distance references (more than two sentences), is accurate. Since they worked with a similar corpus (NY Times articles), we decided to focus only on short distance references.

Although the precision of Lingpipe is high its recall proved to be low (0.54) on our set of documents. It fails to identify some obvious entities and at times cannot associate names of the same person as the co-reference. To improve performance, we made a few modifications to its output as described below.

Sometimes Lingpipe is unable to find the same entity, even when it is repeated exactly the same. For example, if *Albert* is repeated twice in the text, Lingpipe might find only the first occurrence. It is vital for us to find recurrent entities so that we can make the replacements. So we automatically identify and extract any instances of an entity not detected by Lingpipe and assign them the same id that Lingpipe used for that entity elsewhere in the document.

In cases where Lingpipe assigns the same entity repeated in the text different "entity types", we relax the rigid condition that both entities should be person. For example, in one sentence, "Trump" might be a person and in another an organization. However, at least one of them should be person.

To evaluate our algorithm independently of Lingpipe performance, we provide two different types of evaluation: first, by assuming that Lingpipe has detected all the valid entities and second, by taking all of the entities into account whether they are detected by Lingpipe or not. We again use the DUC 2005 documents for our evaluation.

To better explain our results, we will first introduce a few terms. *Action* can be defined as any decision that the algorithm makes. It might be either generating a new pronoun or simply leaving the entity as it is. *Valid Action* is an action acceptable by a human reader. *Invalid Action* is an action unacceptable by a human reader.

Running the algorithm, yields the results summarized in Table 1[5].

Dale (2000) characterizes the mistakes in pronoun generation as missed and inappropriate pronouns. We use the same notation for invalid actions.

Valid Action		Invalid Action	
Replacement	Refusal	Inappropriate	Missed
46	29	7	29
41.4%	26.1%	6.3%	26.1%
67.6%		32.4%	

Table 1. Summary of Algorithm Performance

The number of opportunities for generating a pronoun were 111. As shown in the Table 1, the algorithm works well when deciding to perform a replacement. But, it does not perform well when avoiding (refusing) a replacement. As noted earlier, part of it can be attributed to Lingpipe's failure to extract at least one instance of a repeated entity.

[5] Obviously, the pronouns already in the text are not counted in the evaluation.

Our observation shows that this happens 13 times out of the 29 refusals, meaning that these 13 entities are not found at all, let alone replaced by the pronoun. Table 2 shows the results, omitting these cases in the test data.

Valid Action		Invalid Action	
Replacement	Refusal	Inappropriate	Missed
46	29	7	16
47.0%	29.6%	7.1%	16.3%
76.5%		23.5%	

Table 2. Performance on Alternative Data

Knowing that our experiment concentrated on difficult cases, namely inter-sentential references with no intra-sentence references, the 76.5% accuracy result should be compared with the corresponding result from McCoy et al (1999) which shows an accuracy of 72.6%. An example of a summary produced by our method is provided in Appendix A.

One issue that we did not deal with but which can improve the performance is the identification of appositive clauses in the text. For the sake of brevity in news documents it is very common to use appositives to describe the person's role or job. On the other hand, in English it is not accurate to use a pronoun before an appositive (ex. *He, Canadian Prime Minister*). Hence, replacing the entity before an appositive with a pronoun would become an error.

5. Conclusions and Future Work

In this paper we introduced a simple approach to using an existing co-reference resolution tool in order to perform the task of pronoun generation. The independence from the resolution module enables us to improve the performance with the new advances in anaphora resolution approaches and at the same time enhance the generation module independently.

Since the approach is independent of which anaphora resolution module is used, future work could involve comparisons among different modules. Additionally, competing results from different resolution modules could be scored and combined in order to obtain more accurate generation.

6. References

Gaizauskas R., K. Humphreys. 1996. *Quantitative Evaluation of Coreference Algorithms in an Information Extraction System*. In S. Botley and T. McEnery (eds.), Corpus-based and Computational Approaches to Discourse Anaphora.

Grosz J., Joshi A. K., and Wenstein S.. 1995. *Centering: A Framework for Modeling the Local Coherence of Discourse*, Comput. Linguist., vol.21, no.2,pp.203-225.

Lappin S., H. J. Leass. 1994. *An Algorithm for Pronominal Anaphora Resolution*. Computational Linguistics 20(4):535-561.

McCoy K. F. and Strube M.. 1999. *Generating Anaphoric Expressions: Pronoun or Definite Description?,* Proc. Workshop on the Relation of Discourse/Dialogue Structure and Reference, pp.63-71.

Melli G., Y. Wang, Y. Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar, F. Popowich, 2005, *Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task*, Document Understanding Conference 2005 (DUC-2005), Vancouver, BC.

Mitkov R.. 1998. *Robust Pronoun Resolution with Limited Knowledge.* Proceedings of COLING/ACL 1998, Montreal, Quebec, Canada, 869-875.

Reiter E., R. Dale. 2000. *Building Natural Language Generation Systems*, Cambridge University Press.

Strube M.. 1998. *Never Look Back: An Alternative to Centering.* Proceedings of the 17th International Conference on Comp. Linguist., Montreal, Quebec, Canada, Vol. 2, pp. 1251-1257.

Vicedo J. L, A. Ferrandez. 2000. *Importance of Pronominal Anaphora Resolution in Question Answering Systems.* The 38th Annual Meeting of the Association for Computational Linguistics, ACL 2000.

Appendix A. Sample Output

'There has been a broad recognition, led by Preston, that an institution like the Bank cannot keep on expanding,' says Husain. When Preston came to the Bank **he** found an organisation still shattered by that event. The financial squeeze partly reflects **his** appreciation of the chilly climate. To **his** credit, **he** has no apparent interest in empire building. Having spent 40 years at JP Morgan, the premier New York bank, **he** seems untroubled by the notion of transferring bank functions to the private sector. According to **him**, NGOs have some involvement in 50 per cent of the Bank 's lending activities in Africa. THE World Bank will link loan volume to the strength of a country's efforts to fight poverty, according to an operational directive to staff issued today by Mr Lewis **he**, the bank's president. Mr Barber Conable, the bank's president, says a 50 per cent target for loans directed to the private sector risks 'subterfuge', suggesting the bank would simply redefine loans so they fitted into the right category. In the directive, **he** says poverty reduction is 'the benchmark by which our performance as a development institution will be measured'.