Generation of Referring Expressions in Large Domains

Roman Kutlak **Kees van Deemter** **Chris Mellish**
Department of Computing Science
University of Aberdeen
Aberdeen AB24 3UE, UK
r.kutlak, k.vdeemter, c.mellish @abdn.ac.uk

Abstract

In this article, we investigate generation of referring expressions from large knowledge bases. We discuss some of the issues that arise when using existing referring expression generation algorithms and introduce a corpus-based algorithm that aims to overcome these issues.

The new algorithm is based on the idea of communal common ground and uses a search engine to estimate what properties of a referent are likely to be known by hearers. The algorithm was evaluated against the Incremental Algorithm and human-created descriptions in a hearer-oriented experiment where hearers attempted to identify described people and provide judgements about the used descriptions.

The algorithm outperformed the Incremental Algorithm in terms of the number of correctly identified referents.

Keywords: Referring Expression Generation; Common Ground; Heuristic; Search Engine

Introduction

Any sophisticated Natural Language Generation (NLG) system has to deal with the problem of Referring Expression Generation (REG) (Reiter & Dale, 2000). A number of algorithms have been proposed in the past (see Krahmer and van Deemter (2011) for a survey). These algorithms were often tested on small domains where the number of distractors was often less then ten.

In this article, we are primarily interested in selecting the content for definite descriptions of people in the context of DBpedia[1] (Bizer et al., 2009). Further more, we aim at generation of descriptions that are tailored to hearers. We will also limit our descriptions to do descriptions that do not contain the name of the referent. One reason for the omission of the name is the fact that proper names can often stand on their own. Another reason for omitting the name of the referent are situations where the hearer does not know or recall the name of the referent.

We present a corpus-based algorithm that attempts to rectify some of the issues discussed (avoiding properties that do not help with identification and choosing a good preference order for the Incremental Algorithm). The algorithm was compared with the Incremental Algorithm and human generated descriptions and evaluated in an experiment where human participants judged the quality of the generated descriptions. We conclude the article with a discussion.

[1] DBpedia (dbpedia.org) is an ontology extracted from Wikipedia. The version used for testing the algorithm contained information about more than 1.9 million people.

Related Work

The free availability of DBpedia and other large knowledge bases makes them an interesting resource for NLG. Pacheco, Duboue, and Domínguez (2012) used several REG algorithms to generate referring expressions from DBpedia. Their main focus was on the coverage of DBpedia and the feasibility of using DBpedia in summarisation by re-generation. In particular, Pacheco et al. (2012) used articles from Wikinews[2] and generated referring expressions for the extracted referents (people or organisations) using the Full Brevity(Dale, 1989), Constraint Satisfaction approach (Gardent, 2002) and the Incremental Algorithm (Dale & Reiter, 1995). They found that DBpedia contained information about half of the entities mentioned in the news and that the IA and the Constraint Satisfaction approach were able to generate a definite description in about 98% of the contexts extracted from the news articles. However, the algorithms produced satisfactory definite descriptions only in about 40% of the cases. The two problems identified by Pacheco et al. (2012) were properties that are unique but lead to descriptions of little use to most people and the choice of the preference order of the Incremental Algorithm.

Usefulness of Properties

One of the first computational approaches to determining the usefulness of a property was the use of the *discriminatory power* of a property (Dale, 1989). In order to refer, REG algorithms have to select properties that single out the referent from other entities present in the context in which reference takes place. The other entities that compete for the attention of the hearer are called *distractors*. The discriminatory power of a property is a measure of how many distractors would be removed, should the property be added to a description. The discriminatory power takes a value between 0 and 1. The value of 0 means that the property is true of all entities and thus would not remove any distractors. The discriminator power of 1 means that the property is true of the referent only and thus removes all distractors. Using properties with high discriminatory power leads to short descriptions, which was one of the aims of the early algorithms (Dale, 1989).

[2] wikinews.org

Our experience with DBpedia leads us to believe that the use of discriminatory power might lead to bad descriptions that do not allow the hearer to identify the referent despite of being unique. For example, the property ⟨binomialAuthority : Heritiera percoriacea⟩ means that the person who the property is true of named the species Heritiera percoriacea. Although the description composed of this property is unique, it is unlikely to help the hearer to identify the referent.

To avoid the problem of choosing inappropriate properties, some algorithms (e.g., the Incremental Algorithm and the graph-based approach (Krahmer, van Erk, & Verleg, 2003)) use some sort of pre-determined *preference order*. The preference order is a list of attributes or values [3] that informs the algorithm as to which properties to try first and which properties to avoid when composing a description.

Choosing a Preference Order

The performance of the IA mostly depends on the chosen preference order (van Deemter, Gatt, van der Sluis, & Power, 2012). There are a number of approaches to creating a preference order. For example, one might consult psycholinguistic literature to find out what properties present smaller cognitive load (Pechmann, 1989) or use a corpus to establish what properties are most frequently used (van Deemter et al., 2012).

Probably the most common way of establishing the preference order is to count frequencies of properties in a corpus. In order to do so, the corpus has to be semantically annotated, which is often laborious. Koolen, Krahmer, and Theune (2012) showed that the corpus can be relatively small. We speculate, however, that this will not be the same for large domains as the increased complexity of the domain leads to a larger variation of speakers' expressions. Furthermore, the results obtained by Koolen et al. (2012) apply to a balanced corpus. Most available corpora are not balanced and it is not clear how much data one might need in order to create a good preference order.

Another issue is that it might not always be possible to find a corpus that matches the knowledge base used by the NLG system. NLG systems often start from a particular non-linguistic data and it might not be possible to choose a more suitable data source.

We believe that an algorithm that generates referring expressions in large domains should have the following characteristics:

- The algorithm should not require semantic annotation of a corpus
- Rather than using one preference order, the algorithm should create a preference order for each referent
- The preference order should be created automatically

Having a preference order tailored to each referent should account for differences between individuals and the reasons why hearers know them.

The next section describes our attempt at creating an algorithm for generation of referring expressions in large domains that fulfils the stated criteria.

The Algorithm

Our approach was inspired by the notion of communal common ground (Clark & Marshall, 1981). Clark and Marshall (1981) defined communal common ground as information that is shared by a particular community such as English speakers, Londoners or students attending a particular university. Members of such communities are exposed to certain public information and so they assume that other members of the same group also know such information. Several studies have shown that speakers are sensitive to whether or not the addressee belongs to the same group, and adjust their referring expressions accordingly (Nickerson, Baddeley, & Freeman, 1987; Fussell & Krauss, 1991).

Similarly to the original IA (Dale & Reiter, 1995), our common ground-based algorithm (CG algorithm) treats the attribute `type` differently from other attributes. Our algorithm first selects the most specific value for the attribute `type` and filters out distractors (all entities in DBpedia) that do not have the given type. If any distractors remain, the algorithm examines all properties of the referent and assigns a score to each property using a heuristic based on the notion of communal common ground (described in the next paragraph). The properties are then ordered by the score in descending order. In the manner of the IA, the CG algorithm then takes one property at a time from the ordered list and removes the distractors of which the property is not true. The algorithm terminates when there are no more properties or when the description rules out all distractors. The figure 1 exemplifies the steps taken by the algorithm.

We created a heuristic that uses a set of documents created by a community to estimate what information is likely to be known in the community. Our hypothesis was that if some piece of information is well known in a community, it is likely to be mentioned in a larger number of documents produced by the community. Suppose, for example, that the algorithm is making a choice between two properties of Albert Einstein: ⟨occupation : physicist⟩ and ⟨birthPlace : Ulm⟩. Our heuristic examines the corpus of available documents and counts the number of documents containing the name *Albert Einstein* and *physicist* and the number of documents con-

[3] Most REG algorithms encode the properties of entities as ⟨attribute : value⟩ pairs. When we speak of a property, we mean the pair of an attribute and a value.

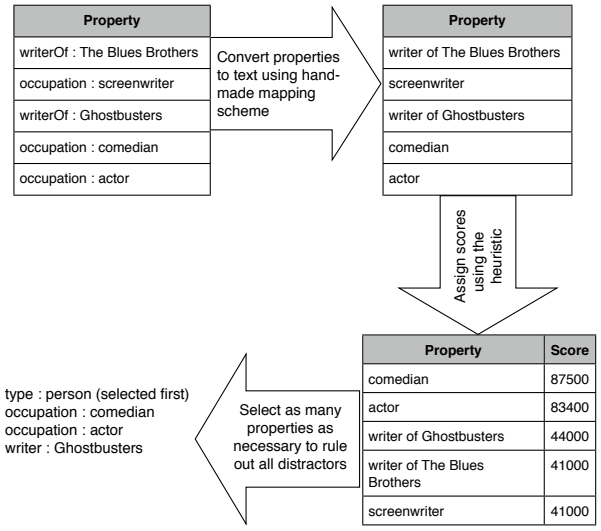

Figure 1: Example of generating description for the referent Dan Aykroyd using the CG algorithm. The selected properties can be surface realised as *This person is an actor and comedian who wrote Ghostbusters.*

taining *Albert Einstein* and *born in Ulm*. If the combination with *physicist* returns more documents than the combination with *Ulm* the heuristic will prefer the property ⟨occupation : physicist⟩ over the less known ⟨birthPlace : Ulm⟩.

The corpus of documents we are using are English websites accessed through the Google search engine. The returned numbers of hits are treated as counts of documents containing the examined property. Search engines have previously been used successfully in several linguistic tasks, such as approximating bigram counts (Keller, Lapata, & Ourioupina, 2002) or measuring word co-occurrence (Turney, 2001).

We tested how well the heuristic correlates with the actual knowledge of people and the heuristic achieved Spearman correlation of 0.64 with $p < 0.001$ (Kutlak, van Deemter, & Mellish, 2012). The correlation with people's knowledge suggests that this heuristic should be able to inform the algorithm as to which properties are more likely to be known by hearers and thus should lead to descriptions that allow better identification.

Unlike the IA, our common ground metric is sensitive to the co-occurrence of the referent's name and each of its properties. This means that the algorithm can order the properties of two referents with the same properties in two different ways (e.g., if both referents are mathematicians as well as physicists, one might be known more as a mathematician and the other more as a physicist). The overall effect is as if the algorithm used preference order tailored to each referent.

The properties in DBpedia are encoded using RDF syntax[4]. In order to be able to use a corpus, the properties had to be converted from their semantic representation to english text. Converting the semantic properties was performed by handmade mapping of attributes to text phrases and appending the corresponding value. Some attributes were mapped to an empty string to achieve more human-like output. For example, the property ⟨occupation : physicist⟩ was converted to *physicist* whereas the property ⟨birthPlace : Ulm⟩ was converted to *born in Ulm*.

Evaluation

Our main focus is on the quality of descriptions from the hearers' perspective, as opposed to modelling speakers. To test the performance of the CG algorithm we performed an evaluation where participants had to guess the name of the described people as well as provide judgements about the descriptions (see section Design for details). This kind of evaluation should provide us with two different views of the algorithms performance. The primary task, where participants had to provide the name of the described person can be seen as an extrinsic evaluation, as it measures the effect of the algorithm on a particular task. The description judgements provide data for intrinsic evaluation where we directly assess some of the properties of the descriptions.

We believe that the CG algorithm's preference for properties that are known by hearers will lead to descriptions that lead to identification more often than descriptions generated by the IA.

- $Hypothesis_1$: The CG algorithm will achieve a higher number of correctly identified referents than the Incremental Algorithm.

We also speculate that both, the IA and the CG algorithm generate descriptions that are too short due to data sparseness and high discriminatory power of certain properties in DBpedia. The length of a description was rated by moving a slider (details in section Design). The slider was initially placed to the middle, which corresponded to the value of 50. Moving the slider to the left indicated that a description was short whereas moving the slider to the right indicated that the description was long. We consider a description to be short if its Length score is below 50.

- $Hypothesis_2$: The descriptions produced by the IA and CG algorithm are short.

We compared the IA and the CG algorithms with human-created descriptions. The human-created description of each referent was taken from the first sentence of

[4]RDF is a W3C standardised language for encoding semantic data. www.w3.org/RDF/

a Wikipedia[5] article describing the referent. This process can be thought of as a simple algorithm that utilises already existing human-produced descriptions. Note, that the primary purpose of the descriptions from Wikipedia is not identification but an overall description of the person. The table 1 shows example descriptions created by each algorithm.

Design

We randomly selected 30 referents from a list of famous people[6]. For each of the 30 famous people we generated a description using the CG algorithm, the IA and using the first line of a Wikipedia entry corresponding to each referent. The preference order used by the IA was derived from the frequencies of properties in an annotated subset of corpus of descriptions of famous people elicited in (Kutlak, van Deemter, & Mellish, 2011). The most frequent attributes (`type`, `occupation`, `nationality`, `...`) were at the beginning of the PO. The DBpedia attributes `name`, `firstName`, `lastName`, `givenName`, `fullName` and `description` were excluded from from the knowledge base to avoid their use.

We used Repeated Latin Square design so that each subject viewed the same number of descriptions generated by each algorithm. The descriptions were randomly assigned to 3 groups to conform with the design. Each participant viewed 10 descriptions created by each algorithm (30 descriptions per participant, one description for each referent).

Procedure

The experiment was conducted online. The participants were given instructions on how to complete the task and were asked to provide some demographic information. Participants then viewed one description at a time and were asked to provide the name of the described person or a reason why they could not provide the name. They had a choice of one of the following three statements:

- I know this person but I cannot recall the name
- The description is adequate but I do not know this person
- The description is inadequate

The participants were also asked to provide ratings for each of the following statements:

- *"How confident are you that you identified the correct person?"*.
  The extremes next to the corresponding slider were "Not at all confident" and "Very confident".
- *"How much information does the description contain?"*.
  The associated extreme values were "Too little" and "Too much".
- *"How likely is it that this description has been produced by a person?"*.
  The associated extremes were "Very unlikely" and "Very likely".

The ratings were provided by moving sliders. The sliders were initially set to the middle and the scale associated with the sliders was 1 to 100 but the values were not visible to the participants.

When participants finished viewing all 30 descriptions they were shown a summary page with the descriptions and corresponding names.

Participants

The experiment was conducted online, and it was distributed through the Amazon Mechanical Turk (MTurk). The experiment was advertised only to the US population of MTurk because the CG heuristic relied on English websites and the stimuli were people famous in the US. Thirty participants (22 females, 8 males) took part in the study, each randomly assigned to one of the three groups. Each of the 90 descriptions was viewed by 10 participants resulting in 900 answes.

Results

Extrinsic Task-based Evaluation From a total of 900 answers, only 285 contained the name of the described person and from those 207 were correct. Table 2 contains more details about the distribution of answers across the algorithms.

The example descriptions in table 1 show one of the problems of the CG algorithm. It sometimes selects properties that "give away" the answer (e.g., the property *the parent of actor Scott Newman*). This problem applies to all three algorithms so we performed another count where descriptions that contain the name of the referent count as **No Guess**. The adjusted counts of correctly identified referents were 60, 23 and 74 for CG algorithm, IA and Wikipedia, respectively.

The numbers of correctly identified referents differed significantly in the case where all correct answers were counted ($\chi^2(2, N = 285) = 22.22, p < 0.001$) as well as in the case where descriptions that "gave away" the answer were removed from the analysis ($\chi^2(2, N = 235) = 19.75, p < 0.001$).

To test $Hypothesis_1$ we performed χ^2 test on the counts of correct and incorrect answers for the IA and the CG algorithm. Our data show that the performance of the CG algorithm was significantly better than the performance of the IA ($\chi^2(1, N = 167) = 20.71, p < 0.001$).

[5] Wikipedia is a community created encyclopaedia. `wikipedia.org`

[6] `http://www.whoismorefamous.com/`

Table 1: Example of descriptions generated by each algorithm. The textual versions of the descriptions produced by the IA and CG algorithm were manually created according to guidelines by a native English speaker who was not involved in the project. This was necessary as the algorithms only select the content of the descriptions (an unordered set of properties).

Referent	Algorithm	Description
Paul Newman	CG	This person was a film director and actor and the parent of actor Scott Newman.
	IA	This person starred in the film Hud and was an entrepreneur.
	Wikipedia	This person was an American actor, film director, entrepreneur, humanitarian, professional racing driver, auto racing team owner and auto racing enthusiast.
Dan Aykroyd	CG	This person is an actor and comedian who wrote Ghostbusters.
	IA	This person is a comedian and starred in Susan's Plan.
	Wikipedia	This person is a Canadian comedian, singer, actor and screenwriter.

Table 2: Number of answers per algorithm. No Guess means that a participant did not fill in the name of the described person. The three categories for No Guess correspond to the three options available to the participants.

			No Guess			
Algorithm	**Correct**	**Incorrect**	**Tip-of-Tongue**	**Unknown Referent**	**Bad Description**	**Total**
CG	90	13	15	121	61	300
IA	35	29	9	172	55	300
Wikipedia	82	36	6	65	111	300
Total	207	78	30	358	227	900

Intrinsic Evaluation As each description was viewed by 10 participants, the final score for each description was the mean value of the 10 scores. Confidence was calculated only on answers where participants provided a name. Table 3 shows the mean Confidence, Length and Humanlikeness ratings for each algorithm. We performed three univariate $3(Algorithm) \times 1(Rating)$ analysis of variance (ANOVAs) on Confidence, Length and Humanlikeness. The analysis showed a significant effect of *Algorithm* on Humanlikeness ($F(2,87) = 4.37, p < 0.05$). The algorithms did not have significant effect on Length or Confidence. The Humanlikeness scores of descriptions extracted from Wikipedia were significantly lower than the scores of descriptions created by the IA and CG algorithms (Post-hoc Tukey's HSD test).

Our second hypothesis was that the descriptions produced by the IA and the CG algorithm were short (score < 50). While this was true for the IA algorithm ($t(299) = -3.417, p < 0.001$) the mean score assigned to the length of descriptions produced by the CG algorithm was not significantly less than 50 ($t(299) = -1.156, p > 0.1$).

Discussion

The CG algorithm outperformed the IA in terms of correctly identified referents. The results in table 2 show surprising number of bad descriptions produced by the Wikipedia "algorithm." This is probably due to the difference of the tasks. While the focus of the evaluation was correct identification of the referent, the primary function of the descriptions from Wikipedia is to provide an over-all description of the person. Such description might be too general (e.g., the description of Sigourney Weaver was *This person is an American actress*). Such descriptions would also explain the why the mean rating of Length of the wikipedia descriptions was so low.

It is also possible that the short Wikipedia descriptions caused the surprisingly low score for Humanlikeness. As the results show, both, the IA and the CG algorithm had significantly higher Humanlikeness ratings than the descriptions from Wikipedia. It seems plausible that when human participants rated the Humanlikeness of a description, they judged not only the likelihood of a person producing such description but also how well the description fulfilled the task. Post-hoc test of the correlation between Length and Humanlikeness showed Pearson correlation $r(88) = 0.61, p < 0.001$.

Conclusion

We have speculated that the existing REG algorithms are not suitable for large domains. The two major problems are the use of *discriminatory power* and the difficulty with creating *preference order* for algorithms that rely on pre-specified order or cost of properties (e.g., IA).

We introduced a novel algorithm based on the notion of communal common ground and evaluated the algorithm against the Incremental Algorithm and human-

Table 3: Mean Confidence , mean Length and mean Humanlikeness (Confidence calculated only on answers where participants provided the name of the referent, Length and Humanlikeness calculated on all answers)

	Confidence		Length		Humanlikeness	
Algorithm	**mean**	**SD**	**mean**	**SD**	**mean**	**SD**
CG	70.97	20.70	48.43	15.57	56.49	12.58
IA	56.76	28.52	45.68	13.25	57.15	11.21
Wiki	55.54	25.45	40.42	22.24	48.73	13.06

produced descriptions extracted from Wikipedia. Our algorithm outperformed the Incremental Algorithm in terms of correctly identified referents and shows to be a promising starting point for further development.

Acknowledgements

We would like to thank Ehud Reiter for his comments on an earlier draft of this article. The research was funded by the Scottish Informatics and Computer Science Alliance (SICSA).

References

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009, September). Dbpedia - a crystallization point for the web of data. *Web Semant.*, *7*(3), 154–165.

Clark, H. H., & Marshall, C. (1981). Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, & I. A. Sag (Eds.), *Elements of discourse understanding* (pp. 10–63). New York: Cambridge University Press.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th annual meeting on ACL* (pp. 68–75). Morristown, NJ, USA: ACL.

Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. In *Cognitive science* (Vol. 19, pp. 233–263).

Fussell, S. R., & Krauss, R. M. (1991). Accuracy and bias in estimates of others' knowledge. *European Journal of Social Psychology*, *21*(5), 445–454.

Gardent, C. (2002). Generating minimal definite descriptions. In *Proceedings of the 40th annual meeting on ACL* (pp. 96–103). Stroudsburg, PA, USA: ACL.

Keller, F., Lapata, M., & Ourioupina, O. (2002). Using the web to overcome data sparseness. In *Proceedings of the acl-02 conference on empirical methods in natural language processing - volume 10* (pp. 230–237). Stroudsburg, PA, USA: ACL.

Koolen, R., Krahmer, E., & Theune, M. (2012, May). Learning preferences for referring expression generation: Effects of domain, language and algorithm. In *INLG 2012* (pp. 3–11). Utica, IL.

Krahmer, E., & van Deemter, K. (2011, 2013/07/21). Computational generation of referring expressions: A survey. *Computational Linguistics*, *38*(1), 173–218.

Krahmer, E., van Erk, S., & Verleg, A. (2003). Graph-based generation of referring expressions. *Comput. Linguist.*, *29*(1), 53–72.

Kutlak, R., van Deemter, K., & Mellish, C. (2011). Audience design in the generation of references to famous people. In L. Carlson, C. Hoelscher, & T. F. Shipley (Eds.), *Proceedings of the 33rd annual meeting of the cognitive science society.* Boston, Massachusetts.

Kutlak, R., van Deemter, K., & Mellish, C. (2012). Corpus-based metrics for assessing communal common ground. *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*.

Nickerson, R. S., Baddeley, A., & Freeman, B. (1987). Are people's estimates of what other people know influenced by what they themselves know? *Acta Psychologica*, *64*(3), 245 - 259.

Pacheco, F., Duboue, P., & Domínguez, M. (2012, June). On the feasibility of open domain referring expression generation using large scale folksonomies. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 641–645). Montréal, Canada: ACL.

Pechmann, T. (1989, 11). Incremental speech production and referential overspecification. *Linguistics*, *27*(1), 89–110.

Reiter, E., & Dale, R. (2000). *Building natural language generation systems*. New York, NY, USA: Cambridge University Press.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the twelfth european conference on machine learning (ecml-2001).*

van Deemter, K., Gatt, A., van der Sluis, I., & Power, R. (2012). Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, *36*(5), 799–836.