Designing Algorithms for Referring with Proper Names

Kees van Deemter
Computing Science Department, University of Aberdeen
k.vdeemter@abdn.ac.uk

Abstract

Standard algorithms for attribute choice in the generation of referring expressions have little to say about the role of Proper Names in referring expressions. We discuss the implications of letting these algorithms produce Proper Names and expressions that have Proper Names as parts.

1 Introduction

Reference – the production and comprehension of referring expressions – has been studied intensively throughout the cognitive sciences. Computational Linguists are no exception, often paying particular attention to the *generation* of referring expressions (REs, (Krahmer and Van Deemter, 2012) for a survey). This area of Natural Language Generation is known as Referring Expressions Generation (REG). An important strand of REG focusses on "one-shot" REs, which do not rely on any linguistic context (precluding anaphoric and other attenuated REs); these are also the primary focus of this paper.[1]

One of the classic algorithm coming out or REG is the Incremental Algorithm (IA) (Dale and Reiter, 1996). Simplifying slightly, the IA starts by ordering properties in a sequence known as the Preference Order. The algorithm starts with an empty RE, then examines the first property from the Preference Order. If this property is true of the referent r and rules out one or more distractors, it is added to the RE; otherwise it is not added, and the next property in the Preference Order is examined. The algorithm terminates when properties $P_{i_1}, .., P_{i_k}$ have been selected that jointly identify the referent (i.e., $[\![P_{i_1}]\!] \cap ... \cap [\![P_{i_k}]\!] = \{r\}$). Different Preference Orders tend to generate different REs, so finding a good one is important.

Proper Names (PNs) are among the most widely studied REs in cognitive science (see e.g., (van Langendonck, 2007), *passim*; (van Deemter, 2016), chapters 2 and 7), and a crucial area of applied work in Information Extraction (e.g., (Jurafsky and Martin, 2009) chapter 22 on Named Entities). Yet REG[2] has neglected PNs, presumably because names could easily trivialise REG: suppose the KB contained a set of people. If only one of the people in the KB is named Obama, then it is easy to identify him, by referring to him by his name. Since PNs tend to make excellent REs, REG would become trivial – so the presumed argument goes.

We argue that this line of reasoning misses some important points and that PNs deserve more attention from researchers in REG.

2 Generating REs that contain a PN

Observe that:

– Name are often ambiguous. "Obama", for instance (not to mention "Smith") could refer to many different people.
– A referent can have many names ("Barack", "Obama", "Barack Obama", etc.) or none.
– A name can combine with other properties and epithets, as in "Mr Barack Obama, America's current president".

[1] See, however, section 2.1 on the use of salience.

[2] An early exception is the *ad hoc* treatment of PNs in (Winograd, 1972)'s SRDLU; recently the possibility of a systematic treatment was suggested as part of (van Deemter, 2014); an exploratory experimental study is (de Oliveira et al., 2015).

Proceedings of The 9th International Natural Language Generation conference, pages 31–35,
Edinburgh, UK, September 5-8 2016. ©2016 Association for Computational Linguistics

– A name can be part of an expression that refers to another referent. The process is recursive, e.g., "The height of the income of Obama's Secretary of State".

So how might PNs be given a place in REG?

2.1 Incorporating Proper Names into REG

Received views of REG suggest that the process contains two steps (Reiter and Dale, 2000): Step 1 decides what general syntactic type of RE to use (e.g., a full description, a PN, a pronoun, or some other type); once this decision is taken, Step 2 (discussed in section 1 above) makes more fine-grained decisions, for example, in case of a full description, this step decides what properties should be expressed in the description. The observations of the previous section make this two-step approach problematic, for example because (in some situations) no PN may be available for a given referent, or because PNs and descriptions must be combined (in other situations). In what follows, we explore a radical alternative, showing that if a suitable representation scheme is used, it is possible to incorporate all decisions related to PNs within Step 2.

Suppose each individual in the KB comes not just with a number of descriptive properties but with 0 or more PNs as well, where a PN is regarded as a property that is true of all individuals who bear this name.

– (being named) Joe Klein is a property of all individuals named Joe Klein
– (being named) Joe is a property of all those individuals named Joe
– (being named) Klein is a property of all those individuals named Klein

The idea that a PN can be viewed as a property of its bearer deviates from a long tradition of work in philosophy and logic that regards PNs as *rigid designators* (Kripke, 1980), yet it enjoys considerable support. (Burge, 1973), for example, observes that PNs can behave like common nouns, as in "There are relatively few Alfreds in P", and "An Alfred joined the club today" (see (Larson and Segal, 1995) and (Elbourne, 2005) for further support).

A simple KB containing PNs as well as ordinary properties could look like this:

JOB: political commentator, commentator
NATIONALITY: American
NAMES: Mr Joe Klein, Joe Klein, Joe, Klein

Because longer versions of a person's name are applicable to only some of the individuals to whom a shorter version is applicable, the values of the NAMES attribute often *subsume* each other: all people who are called Mr Joe Klein are also called Joe Klein, and so on. These properties can be dealt with using the mechanism for subsumption in the Incremental Algorithm (which would also state that all *political commentators* are *commentators*, for instance) (Dale and Reiter, 1996).

Of course if Joe Klein is the only Joe in the room, we can refer unambiguously to him saying "Joe". This is accounted for by making the REG algorithm that operates on the KB above salience aware in one of the standard ways, e.g., (Krahmer and Theune, 2002). Salience also suggests a way in which REG can extend beyond one-shot REs to cover reference in extended discourse or dialogue: if x is introduced by means of the PN "Joe Klein" in a text, then if x is the only Joe so far mentioned, then this makes x the most salient of all Joe's, licencing the short RE "Joe".

In short:

– Each object has an attribute NAMES.
– The set of values of NAMES can be empty (no name is available), singleton (one name), or neither (several names).
– A subsumption (i.e., subset) relation can be defined among these values.
– Different objects can share some or all of their names.

If names are the "canonical" way of referring to an entity, then standard mechanisms could be invoked to favour names at the expense of other properties. One option is to Dale and Reiter's Preference Order (Dale and Reiter, 1996), making NAMES the most highly preferred attribute in an Incremental Algorithm. Alternatively, a new type of brevity-based algorithm might be used that generates the RE that contains the smallest number of *syllables*.[3] Assuming that PNs are brief (as they often are), this type of approach would favour PNs, and it would favour shorter PNs over longer ones (e.g., "Klein" over "Joe Klein"). It would also predict that PNs are avoided

[3] Note that this approach would measure brevity as a surface property of a string, unlike the Full Brevity algorithm of (Dale, 1989), which sees brevity as a semantic property, letting REG choose the RE composed by the smallest number of *properties*.

where large sets are enumerated (compare the RE "the citizens of China" with an enumeration of all the elements of this set).

To see how REG could work in an Incremental Algorithm, consider a simple KB, where each individual has 1 name:

TYPE: woman $\{w_1, w_2, w_3\}$, man $\{m1\}$, dog $\{d_1, d_2\}$
NAMES: mary $\{w_1\}$, shona $\{w_2, w_3\}$, rover $\{d_1\}$, max $\{m_1, d_2\}$
ACTION: feed $\{(w_1, d_1), (w_2, d_2), (w_2, d_1)\}$
AFFECTION: love $\{(w_1, d_1), (w_3, d_1)\}$

This approach generates REs such as:

d_1: "Rover"
d_2: "The dog called Max"
w_3: "Shona, who loves a dog"

With the above representation scheme in place, classic REG algorithms can be applied without modifications. However, the scheme does not allow PNs to have properties (e.g., "is a posh name", "has 5 characters" ,"is common in Scotland"). If names are *reified*, then this becomes possible; what's more, PNs themselves could be referred to (e.g., "the name his friends call him"): a name is just another object linked (on the one hand) to the things it names and (on the other hand) to the ways in which it manifests itself in spelling, pronunciation, etc. For example, n_2 may name both a man and a dog, and it may be written as "Max":

Type: woman $\{w_1, w_2, w_3\}$, man $\{m_1\}$, dog $\{d_1, d_2\}$, name $\{n_1, n_2, n_3, n_4\}$
Action: feed $\{(w_1, d_1), (w_2, d_2), (w_2, d_1)\}$
Affection: love $\{(w_1, d_1), (w_3, d_1)\}$
Naming: name $\{(d_1, n_1), (d_2, n_2), (w_1, n_3), (w_2, n_4), (w_3, n_4), (m_1, n_2)\}$
Spelling: written $\{(n_1, Rover), (n_2, Max), (n_3, Mary), (n_4, Shona)\}$

Standard REG algorithms can use this KB to generate "The name shared by a man and a dog" (i.e., "Max"). If n_4 is Scottish, we obtain "women with a Scottish name" as well. A slight drawback of this approach, which treats names as objects, is that subsumption can no longer be used to compare names.

2.2 Challenges facing this approach

This approach works, but it puts a spotlight on some difficult issues, some of which affect the generation of *descriptive* REs as well:

1. PNs are *not* always preferred. For example, if the Director of Taxes is Mrs X, this does not mean that "Contact the Director of Taxes" is always better worded as "Contact Mrs X", since her job title may be relevant. The lack of a computational theory of *relevance* affects all of REG but becomes very noticeable in the choice between PNs and descriptions.

2. There is no reason for limiting reification to PNs. Colours too could be reified, for example, to generate "the colour of grass". The traditional dichotomy between objects and properties limits the range of REs that these algorithms can generate.

3. REG algorithms are ignorant about social relations between speaker, hearer, and referent. Consider a couple with a son and a daughter. Speaking to his mother, the son could say "my sister", "your daughter", etc., yet in most situations a PN would be better. Titles and epithets like "Dr" and "Aunt(y)", complicate matters further.

4. As elsewhere in REG, questions about overspecification need to be faced. When, for example, is it useful to add an *appositive* to a PN, as in "Mr Barack Obama, *America's current president*"? Furthermore, Linguistic Realisation will have to decide about the surface order of the PN and the appositive, perhaps depending on whether the PN and/or the appositive (by itself) refers uniquely.

5. If PNs are properties of the referent, then this leaves room for expressing one and the same PN with a different string. (For example, "Doctor" may be worded as "Doctor", "Dr.", or "Dr".) The desirability of this use of Linguistic Realisation would need to be investigated.

6. It is often difficult for the speaker to assess whether the hearer knows who a given PN refers to. The hearer may never have heard of Joe Klein, for example, and this would cause the RE "Joe Klein" to mis-fire. Lack of shared knowledge is a problem for *descriptive* REs as well, but it is exacerbated in the case of PNs, because names are highly conventional: once I've learned what "red" means, I can apply the word to any red object, but learning your name does not teach me to apply this name to anyone else.

The last point has important implications. Imagine a programmer wanting to implement the algorithm of section 2.1, aiming to mimic human language use. If she decides to implement an Incremental Algorithm, then how to choose its free pa-

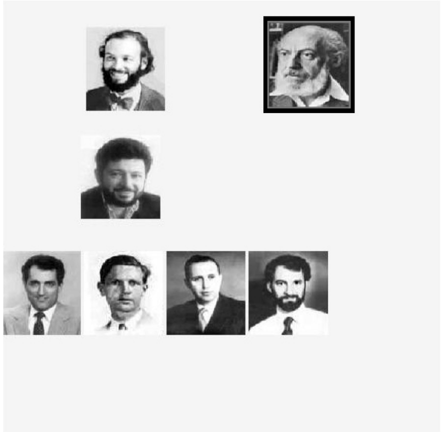

Figure 1: *A trial in the "people" part of the* TUNA *experiment*

rameter, the Preference Order? She could learn one via an elicitation experiment, but how does she find a generic REG algorithm that works for all PNs?

Consider a scene from an experiment where speakers referred to stimuli on a screen (van Deemter et al., 2012). Participants called the man in the top right "the man with the white beard", etc. They *might* have said "Samuel Eilenberg", yet no-one did, because participants didn't know his name. Participants could have been trained to be familiar with every individual's name, but this could easily have *primed* the use of names at the expense of descriptions; the same happens when names are visible as captions, as was done in (de Oliveira et al., 2015) using fictitious names of geographical areas; see also (Anderson et al., 1991). Such an approach does not give reliable information on how REG algorithms should choose between PNs and descriptions. The problem is not just that PNs are conventional, but that their conventional meaning can be entrenched to different degrees, varying from short-lived "conceptual pacts" (Brennan and Clark, 1996) to names that are very widely known and used.

2.3 Lessons from situations where PNs are avoided

Suppose someone asks "Who is Joe Klein?" (cf., section 2.2, point 6). Would it make sense to respond "(He is) the author of the bestselling political novel of the 1990s?" It depends on the importance of this fact and how widely it is known.

To model answers to "Who is?" questions (see (Boër and Lycan, 1986) for a theoretical study), (Kutlak et al., 2013) designed a REG algorithm that employs the following Heuristic: Based on the frequency with which a name n co-occurs with a property P, the Heuristic estimates how likely the proposition $P(n)$ is to be known by an arbitrarily chosen hearer. Evaluation studies suggest that this Heuristic goes a long way towards estimating how many people know a fact, and the complete REG algorithm (which involves 2 other heuristics) outperforms its competitors in terms of its ability to generate descriptions that allow hearers to guess correctly the name of the referent. Although the authors focussed on the WWW, the approach can use any corpus that represents the ideas of a community (e.g., a company's intranet).

This approach suggests a promising handle on the conventionality of PNs. It allows us to estimate, for example, the likelihood that a name like "Joe Klein" is known by hearers to refer to the commentator and novelist of that name, and this would allow us to limit the KB of section 2 to names that are well enough known. We hypothesise that PNs have a *higher likelihood* of being uttered as part of REs by members of a community (e.g., users of the WWW) the more frequently these PNs occur as names of this referent in documents produced by that community. Further experiments could flesh out how the use of PNs depends on a number of factors, including the Knowledge Heuristic. Essentially, PNs would be treated as properties of a referent that may or may not be known to the hearer, analogous to the descriptive properties of (Kutlak et al., 2013).

3 Conclusion

We have shown how, given appropriate semantic representations, standard attribute algorithms are able to generate REs that contain PNs, thereby solving problems with the standard 2-step perspective on REG that separates choosing the general syntactic type of RE from more fine-grained decisions about the content of the RE. However, our approach raises difficult questions about the choices that a REG algorithm needs to make between PNs and descriptive REs. We argue that some of the trickiest questions in this area may be solved if large corpora are employed as a source of insight into the degree to which a PN is likely to be known by the recipient of the RE.

References

Anne A. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. 1991. The HCRC map task corpus. *Language and Speech*, 34:351–366.

Steven E. Boër and William G. Lycan. 1986. *Knowing Who*. MIT Press, Cambridge, Mass.

Susan Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493.

Tyler Burge. 1973. Reference and proper names. *The Journal of Philosophy*, 70:425–439.

Robert Dale and Ehud Reiter. 1996. The role of the gricean maxims in the generation of referring expressions. In *AAAI–96 Spring Symposium on Computational Models of Conversational Implicature*.

Robert Dale. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 68–75.

Rodrigo de Oliveira, Somayajulu Sripada, and Ehud Reiter. 2015. Designing an algorithm for generating named spatial references. In *Proceedings of 15th European Workshop on Natural Language Generation (ENLG-2015)*, pages 127–135, Brighton, UK.

Paul Elbourne. 2005. *Situations and Individuals*. MIT Press, Cambridge, Mass.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (second edition)*. Pearson, Upper Saddle River, NJ.

Emiel Krahmer and Mariët Theune. 2002. Efficient context–sensitive generation of descriptions in context. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264, CSLI Publications, CSLI, Stanford.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: a survey. *Computational Linguistics*, 38(1):173–218.

Saul Kripke. 1980. *Naming and Necessity*. Harvard University Press, Cambridge, Mass.

Roman Kutlak, Kees van Deemter, and Chris Mellish. 2013. Generation of referring expressions in large domains. In *Proceedings of the workshop Production of Referring Expressions, associated with the 35th Meeting of the Cognitive Science Society*.

Richard Larson and Gabriel Segal. 1995. *Knowledge and Meaning. An Introduction to Semantic Theory.* MIT Press, Cambridge, Mass.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Kees van Deemter, Albert Gatt, Ielka van der Sluis, and Richard Power. 2012. Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, 36(5):799–836.

Kees van Deemter. 2014. Referability. In A. Stent and S. Bangalore, editors, *Natural Language Genetration in Interactive Systems*, pages 95–125. Cambridge University Press.

Kees van Deemter. 2016. *Computational Models of Referring: a Study in Cognitive Science*. MIT Press, May 2016.

Willy van Langendonck. 2007. *Theory and Typology of Proper Names*. Mouton de Gruyter, The Hague.

Terry Winograd. 1972. *Understanding Natural Language*. Academic Press, New York.