Sentence Trimming in Service of Verb Phrase Ellipsis Resolution

Marjorie McShane (margemc34@gmail.com)
Sergei Nirenburg (zavedomo@gmail.com)
Petr Babkin (petr.a.babkin@gmail.com)
Cognitive Science Department, Rensselaer Polytechnic Institute
110 8th Street, Troy, NY, USA

Abstract

We describe two methods of improving the coverage of a system that automatically detects and resolves verb phrase ellipsis. Both methods involve recognizing non-core sentence constituents, thereby making the core constituents more easily manipulated by the ellipsis detection and resolution functions. A system evaluation shows increases both in the number of sentences in which ellipsis is detected, and in the percentage of elliptical sentences that can be treated by the system's methods.

Keywords: ellipsis; VP ellipsis; natural language processing; sentence trimming; syntactic pruning

Introduction

Ellipsis is defined as the non-expression of linguistic material that can be reconstructed by the interlocutor. The work reported here focuses on detecting and resolving verb phrase (VP) ellipsis that is licensed by a modal or auxiliary verb. For example, in (1) the modal verb *can* licenses ellipsis of the meaning of its scope, *get bragging rights*. (Elided categories are indicated by [e]; their sponsors – typically, antecedents – are indicated in italics.)[1]

(1) And you try to *get bragging rights* if you can [e].

McShane and Babkin (2015) report a VP ellipsis resolution system that is novel in three ways. First, NLP (natural language processing) systems tend not to treat many kinds of ellipsis since it is resistant to the currently dominant method of supervised machine learning, which relies on annotations of visible (not elided) text strings. Second, our development methodology is knowledge-based, leveraging human-oriented linguistic insights as heuristic evidence. In essence, we are trying to teach the machine to do *what* people do by modeling (to some degree) *how* people seem to do it. This places the work squarely in the paradigm of AI-NLP (artificial-intelligence-inspired NLP). Third, since both detecting and resolving ellipsis are difficult problems, the system is configured to independently select which examples it believes it can treat with reasonably high precision, and treat only those.

This partial-coverage approach has potential benefits for two communities. For mainstream NLP, treating at least some elided VPs is preferable to not treating any. For the intelligent agent community, we believe it is essential for agents to be able to judge their own confidence in all aspects of language processing, then use those confidence estimates to guide their next move. So, in cases of high confidence in language analysis, the system can boldly proceed to decision-making and action, whereas in cases of low confidence, it should seek clarification from its human collaborator.

Although the initial evaluation of our system (McShane and Babkin 2015) was promising, one area for improvement was low coverage of examples, both with respect to detecting ellipsis and with respect to selecting which examples to resolve. Both of these are improved upon in the enhanced system reported here. However, to understand the nature of the improvements, one must first understand the basics of the original system.

Detection of VP ellipsis was carried out very simply: any modal or auxiliary verb directly preceding a hard discourse break – defined as a period, semi-colon or colon – was considered an ellipsis licensor (cf. (1)). The reason for orienting around hard discourse breaks was practical: for our initial system development, we sought a cheap, fast method of finding elliptical examples in a large corpus without too many false positives. Although this method did offer high precision, it had less than optimal recall.

In the new version of the system, we expand the detection heuristics to also include modal and auxiliary verbs occurring before a *soft* discourse break, defined as a comma, dash, or open parenthesis. However, this detection heuristic is more error-prone because "[modal] + [soft discourse break]" does not always signal ellipsis: the modal's complement can actually occur later on in the sentence. E.g., in (2) the scope of *tried to* is *check with other several sources*.

(2) "I've always tried to, when we get intelligence, check with other several sources, ..."

To weed out false positives, we implemented parenthetical detection functions that attempt to determine the role of each soft discourse break that follows a modal or auxiliary verb. The punctuation mark could either (a) introduce a parenthetical that is then followed by the scope

[1] All cited examples except for (4), (22a) and (23a) – which were invented – are from the Gigaword corpus (Graff and Cieri 2003), which was used for system evaluation. Both the Gigaword corpus and the COCA corpus (Davies 2008-) were used for linguistic analysis.

of the modal/auxiliary (i.e., there is no VP ellipsis) or (b) not introduce a parenthetical, in which case the structure is likely elliptical. To summarize, *the first advancement reported here is the use of parenthetical detection strategies that permit the system to detect ellipsis before soft discourse breaks; this increases system coverage at the stage of ellipsis detection.*

As concerns resolution, the system attempts to resolve only those cases of ellipsis that it believes it can treat with reasonable confidence. Below we briefly describe two of its resolution strategies.

Pattern matching. We have recorded nine broadly-defined phrasal patterns (which divide into many more subpatterns) that include VP ellipsis, along with their ellipsis resolution strategies. For example, (3) matched the pattern *what NP *can*[2] and the system correctly indicated that the sponsor was *say*.

(3) Vincent Schmid, the vicar of the cathedral, said prayer and music would *say* what words could not [e].

We will not detail the pattern-matching strategy here, since we have no enhancements to report; however, it is important to understand that pattern matching is the first ellipsis resolution strategy to fire, and it takes care of many cases of VP ellipsis.

The Simple Parallel Configuration. Another strategy for treating VP ellipsis is to identify contexts that we call Simple Parallel Configurations, which are structurally simple enough to be treated without the need for deep reasoning or world knowledge. We operationalized the notion of Simple Parallel Configuration in terms of Stanford CoreNLP (Manning et al. 2014) dependency parses. Configurations are deemed Simple Parallel if they contain:

- exactly one instance of a "whitelisted" dependency – i.e., a *conj, advcl* or *parataxis* dependency that links the modal/auxiliary element licensing the ellipsis with an element from the sponsor clause;[3]
- no instances of a "blacklisted dependency'" – i.e., a *ccomp, rcmod, dep* or *complm* dependency, all of which indicate various types of embedded verbal structures that complicate matters by offering competing candidate sponsors;
- one or more instances of a "gray-listed" dependency, defined as an *xcomp* or *aux* dependency that takes as its arguments matrix and/or main verbs from the sponsor clause.

For example, the parse for (4) includes one whitelisted dependency, **conj**(wanted-2, did-10), and three gray-listed dependencies – **xcomp**(wanted-2, try-4), **xcomp**(try-4, start-6), **xcomp**(start-6, juggle-8).

(4) John wanted to *try to start to juggle* and did [e].

Once the system detects a Simple Parallel Configuration, it still needs to resolve the ellipsis. Here, the decision space can be complex. Although the whitelisted dependency indicates which *clause* contains the sponsor, the system still must determine which elements from that clause should participate in the resolution: e.g., are modal verbs and adverbs part of the sponsor or not? (For example, in (4) the leftmost member of the sponsor might be interpreted as *try* or *start*). In the reported evaluation, the system is responsible for selecting only the correct verbal head of the sponsoring VP. So, whereas it *is* responsible for decisions about including/excluding modal verbs like *want to, try to,* and *start to* in (4), it is not responsible for decisions about other non-head elements, such as adverbs.

Orienting around Simple Parallel Configurations captures the intuition that some elliptical contexts are quite simple and straightforward, whereas others are not. It makes sense to prepare agents to resolve the simpler cases in the near term as we work toward conquering the more difficult cases over time.

Making more contexts look Simple Parallel. Some elliptical sentences that are not Simple Parallel are truly difficult. For example, (5) offers several competing candidate sponsors and requires both world knowledge and close attention by a human to resolve the ellipsis.

(5) The former Massachusetts governor called on United Nations Secretary General Ban Ki-moon to *revoke* Ahmadinejad's invitation to the assembly and warned Washington should reconsider support for the world body if he did not [e].

Our system does not currently attempt to treat contexts like these.

But other non-Simple Parallel examples look very much like Simple Parallel Configurations if only some parts were omitted. For example, the boldface portion of (6) would be very straightforward for ellipsis resolution if only the portion formatted using strikethrough would disappear (the portion after the quoted speech is irrelevant for the process of ellipsis resolution).

(6) "~~We're celebrating the fact that we're living in a time where~~, **when we want to *be in the kitchen*, we can [e]**," says Tamara Cohen, Ma'yan program director.

This leads us to *the second advancement reported here, which is the use of sentence trimming strategies that permit the system to transform complex sentences into simpler ones that can be treated as Simple Parallel Configurations.* Sentence trimming follows the psychologically motivated

[2] The asterisk indicates any inflectional form of this verb or select related verbs.

[3] *Conj* dependencies that take non-verbal arguments are ignored, since they can reflect, e.g., nominal conjunction structures such as *Lulu and Fido*. Definitions of the dependencies can be found in Stanford CoreNLP dependencies manual, found here: http://nlp.stanford.edu/software/dependencies_manual.pdf.

hypothesis that some sentence constituents are more salient to the meaning of the utterance than others. Focusing on the core ones can have useful side-effects for the difficult task of automatic ellipsis resolution.

Of course, parenthetical detection can be framed as a subclass of sentence trimming, since one way to trim a sentence is to detect and remove parenthetical information. However, since parenthetical detection and overall sentence trimming are exploited at different points and to different ends in the system, we treat them separately in the narrative below.

Parenthetical Detection

To reiterate, when we expanded our ellipsis detection method to include the detection of elided VPs before *soft* discourse breaks (in addition to hard discourse breaks), we had to introduce a parenthetical detection strategy to avoid false positives. This strategy operates on the output of Stanford CoreNLP parsing and is comprised of 12 functions, presented below with examples. Note that one cannot rely on paired punctuation marks to delineate parentheticals since one or both punctuation marks can be omitted.

1. The **prn** dependency in the Stanford CoreNLP parse detects some cases of parentheticals directly: *, they wondered,*
2. Conjunction + ($NP_{SUBJECT}$) + modal verb: *and did, and need not, or wishes to, and one should not*
3. Prepositional phrase: *among others, at any price*
4. Adverb: *however, therefore, potentially*
5. Adverbial phrase: *absolutely not, more than ever*
6. Conjunction + clause: *as he put it, as you know*
7. (Quasi-)Idiom: *as is the case/situation with*
8. Conjunction + subjectless past-participial clause: *if untreated, as previously announced, if given in sufficient doses, if needed, as put so eloquently*
9. Conjunction + adjective: *if possible*
10. Clause without object or complement: *it seems, you know, NP_{SUBJ} feel <believe, imagine, think, guess, hope, etc.>*
11. Gerund phrase: *gritting our teeth, following a review*
12. Two modals "share: a scope, both appearing elliptical at the surface but having a textual postcedent, as shown in (7).

(7) "The possibility for events to spiral rapidly out of control in circumstances of darkness, high emotions, low trust and official uncertainty **cannot, and should not, be underestimated,**" DeGolyer said in a report published last July.

When the system detects "[modal/aux.] + [soft discourse break] + [parenthetical]", it considers the context to be non-elliptical since the scope of the modal/aux. generally follows the parenthetical. In all other cases, the soft discourse break is treated as if it were a hard discourse break: an elided VP is posited after the modal and the post-punctuation portion of the sentence is disregarded for subsequent processing.

Sentence Trimming

To simplify complex sentences into, ideally, Simple Parallel Configurations, we implemented 7 sentence trimming procedures, which rely on the output of Stanford CoreNLP parsing. The procedures can, individually or in combination, transform a complex context into one that can be treated as a Simple Parallel Configuration. We briefly describe each trimming strategy in turn. Illustrative examples indicate the trimmed part using strikethrough.

1. Strip sentence adverbs. We created a list of over 500 sentence adverbs, based on a combination of introspection and searches using the online version of the COCA corpus (Davies 2008-).[4]

(8) ~~Even after that~~ I was thinking about *sprinting* and being in front, but I could not [e].

2. Strip pre-punctuation clause. The system walks backwards through the text. If it encounters a comma, dash, semi-colon or colon, it strips it off along with the preceding context. If the remaining portion is a Simple Parallel configuration, it resolves the ellipsis. If not, it continues walking back through the text to the next punctuation mark.

(9) ~~I was OK,~~ I tried to *find* my game but I couldn't [e].

3. Strip speech/thought verb and preceding context. The system walks backwards through the text. If it encounters one of a listed inventory of speech/thought verbs, it removes that verb and all preceding content and evaluates whether the remaining structure is Simple Parallel. If it is, the system resolves the ellipsis.

(10) ~~Barak told Israel TV that the agents asked if~~ he would *help* them in their investigation of the attacks if he could [e].

4. Strip pre-conjunction material. The system walks backwards through the text to the first encountered conjunction. If it is among our listed 28 conjunctions, and if the associated dependency takes verbal arguments, then the system determines whether the latter conjunct is a Simple Parallel configuration. If yes, the system resolves the ellipsis. If not, it continues to walk back through the text to determine if adding another conjunct will result in a Simple Parallel Configuration.

For example, when encountering *and* in (11) the system evaluates whether *I couldn't* is Simple Parallel: it is not. So the system continues walking back to the next conjunction,

[4] For example, we searched for frequent single words, and 2- and 3-word collocations, occurring between a period and a comma.

because, and prunes off the text prior to it. Since what remains is a Simple Parallel Configuration, the system resolves the ellipsis.

(11) ~~My legs make the serve because~~ you need to *bend* your knees and I couldn't [e].[5]

5. Strip sentence-initial PPs and adverbs. These are detected from the parse tree.

(12) ~~In the swimming test~~, inosine-treated rats by week eight were able to properly *control* their forepaws, while the untreated rats could not [e].

6. Strip parentheticals. The approach to stripping parentheticals is essentially the same as described earlier; however, in this case, the parenthetical need not be preceded by "[modal/aux. verb] + [soft discourse break]".

(13) By winning a second term, Bush has *accomplished* what his father – ~~defeated in 1992 by Democrat Bill Clinton~~ – could not [e].

7. Strip non-quotative *NP said/was told,* etc. The collocations *NP said, NP was told* and paraphrases thereof are often inserted into propositions that are not direct quotes, as in (14).

(14) Belu said he wanted to *protest*, but ~~was told~~ he could not [e].

Evaluation

This evaluation measured the changes in the coverage of elliptical examples due to the enhancements described above, and also measured the precision of resolution for all experimental runs. Evaluation was carried out on a random sample of the Gigaword Corpus (Graff and Cieri 2003). It must be noted that samples of this same corpus were used for linguistic investigation of component phenomena and testing of algorithms – i.e., before engaging in development work, we did not set aside a dedicated evaluation segment. However, we believe the evaluation results are still valid since this is a very large corpus and we did not seek to tune our approach to cover any individual examples.

We carried out two phases of evaluation. Phase 1 focused primarily on the effects of trimming procedures. First we semi-automatically – i.e., automatically followed by manual checking – identified examples of VP ellipsis before a hard discourse break (HDB) and before a soft discourse break (SDB). We then ran the Simple Parallel Configuration detector over those examples to determine how many it could treat. Column 3 of Table 1 shows the number of actually elliptical examples that were evaluated for both HDB and SDB contexts. The Simple Parallel column indicates how many of the examples were treated as Simple Parallel Configurations, without trimming and with trimming (Column 2 indicates whether trimming was applied). Recall indicates this number of examples treated as a percentage of total examples. Head precision refers to accuracy of detecting the correct head of the sponsor.

Table 1. Evaluation of sentences that were confirmed to be elliptical.

DB	Trim	Elliptical Examples	Simple Parallel	Recall	Head Precision
hard	no	105	28	27%	71%
	yes		48	46%	71%
soft	no	109	13	12%	77%
	yes		20	18%	75%

Without trimming, the system treated 28/105 HDB examples (27%) and 13/109 SDB examples (12%). Next we applied trimming procedures to the untreated sentences, which increased recall to 48/105 (46%) for HDB examples and 20/109 (18%) for SDB examples. Resolution accuracy was about the same with and without trimming.

Phase 2 of the evaluation observes the system in fully automatic mode: i.e., we did not manually verify that the extracted examples actually were elliptical. Table 2 shows the percentage of examples the system could treat under each of the four experimental conditions *as well as* the number of examples treated by our inventory of elliptical phrasal patterns, which were run before the Simple Parallel engine was launched. Although our pattern-based methods were not described in depth in this paper, this count helps to convey the relative proportion that each system module contributes to the overall goal of resolving VP ellipsis.

Table 2. Evaluation of the system in fully automatic mode, from detection through resolution.

DB	Trim	Examples	Simple Parallel	Head Precision
phrasals	N/A	150	N/A	83%
hard	no	95	13	77%
	yes		32	72%
soft	no	144	23	78%
	yes		31	71%

Note that Table 2 does not include a Recall column – instead, we orient around how many of the examples that the system thought were elliptical could be treated by our methods, and what percentage of those resolved were resolved correctly. The reason for not including a formal measure of "recall" is that there is no clean, useful definition of that in this system configuration, since there can be false positives at the extraction stage. The system should not be penalized for failing to resolve an instance of "ellipsis" that was actually never ellipsis to begin with. Moreover, some of the contexts in this corpus were essentially word salad,

[5] The fact that the resolution requires sloppy identity of the object – i.e., *bend MY knees* – will not be treated in this paper.

uninterpretable even by people. If the system chose not to treat such sentences, that was appropriate.

Interpretation of Evaluation Results

Orienting evaluation strictly around numbers does not convey the full picture for knowledge-based systems, where error analysis is key to improvements. So let us give just a brief taste of what that process revealed.

First, we should emphasize that the system arrived at many impressive results, such as its correct treatment of examples (15)-(18).

(15) "We have shown that we can play exciting football and should have had that game won but you just can not afford to *switch* off for even a second and I am afraid we did [e].

(16) Airline analysts said the Mesa Air initiative may have prompted Northwest, which already owns a large chunk of Mesaba and has executives on its board of directors, to *jump* in with an offer before Mesa did [e].

(17) Prosecutors say they try to avoid *calling* journalists to testify, but sometimes they must [e].

(18) "If we must [e], we can *allow* 80 or 100 officers to retire, on condition that they be replaced by officers capable of leading an army."

Sentences (15) and (16) include many candidate sponsors to be selected from. Sentence (17) requires the system to strip *try to avoid* from the sponsor, leaving *calling* as the head of the ellipsis resolution. And sentence (18) requires the system to find a postcedent, rather than the more typical antecedent (this resolution strategy is formulated as a phrasal pattern).

One source of errors, which is the focus of ongoing work, is the treatment of structurally embedded categories: e.g., in (19) the system selected *capable* (underlined) as the head of the sponsor rather than its complement, *increasing*; and in (20) it should have stripped *would not* from the actual sponsor, *happen*.

(19) Khelil, speaking in an interview with OPECNA, said he was not sure the members of OPEC were capable of easily *increasing* production, even if they wanted to [e].

(20) They said the elections would not *happen*, and they did [e].

Another common error involves cases in which the actual antecedent is not within the given sentence, but the given sentence contains what appears to be a valid sponsor.

(21) "But I feel good that if I need to [e], I will."

In some cases, our structurally-oriented rules misfire for reasons that can only be understood with the help of semantic analysis. For example, in (22) the actual sponsor is in the preceding context; but if we slightly edit the sentence to the form in (22a), our rule would have fired correctly.

(22) "Even if we can [e], we can't afford it."

(22a) "Even if we want to [e], we can't *buy* it."

A similar understandable but incorrect resolution occurred in (23). (23a) is a structurally similar context in which the system's resolution would have been appropriate.

(23) He appealed to Indonesians to *respect* national stability and threatened to call out the army if they did not [e].

(23a) He threatened to *call* out the army if they did not [e].

Returning to the big picture, this system is being tasked with a difficult challenge: it must both detect and resolve ellipsis; it takes as input sentences that might be non-normative or semantically difficult; and it uses as parse that, naturally, can include unexpected results. This is a problem space that has been undertreated in computer systems to date, and we believe that the approaches we have described here are a strong first step.

Related Work

One relevant related work on VP ellipsis is Hardt's (1997) VP ellipsis system. However, whereas that system requires a perfect (manually corrected) syntactic parse, ours uses the results of automatic parsing.

Extensive work has been devoted to the automatic resolution of overt referring expressions, with a recent notable contribution being Lee et al. (2013).

As concerns sentence trimming, much of the past work has been in service of text summarization. For example, Knight and Marcu (2002) implement two approaches to sentence compression (a noisy-channel, probabilistic approach, and a decision-tree, deterministic one) using a methodology that involves aligning sentences from a source document (called 'Text') with sentences from manually generated abstracts of the document (called 'Abstract'), then using these <Abstract, Text> tuples to learn how to trim Texts into Abstracts. Gagnon and Da Sylva (2005) trim sentences based on a dependency parse, removing subtrees that represent certain types of relations, such as prepositional complements of the verb, subordinate clauses and noun appositions. Apart from summarization, sentence trimming has been applied to headline generation, event extraction and subtitling. Zajic et al.'s (2004) Hedge Trimmer system produced headlines by compressing the lead sentence of an article and removing constituents (articles, prepositional phrases, auxiliary *have/be,* etc.) in a particular order until the desired length threshold was reached. Buyko et al.'s (2011) trimmer supported event extraction by pruning what they call "informationally irrelevant lexical material" (such as auxiliary and modal verbs) from dependency graphs in order to focus on semantically rich dependencies.

Perhaps the closest precedent to our approach is the one reported in Vanderwende et al. (2007), which involves 5 trimming patterns. Three directly trim nodes generated by the parser (noun appositive, gerund clause, nonrestrictive relative clause). The fourth pattern is the deletion of lead conjunctions and adverbials (of time and manner only), which relies on a parser feature indicating time/manner adverbials. The final pattern, intra-sentential attribution (e.g., "…the report said that…") requires direct manipulation of the parse. Interestingly enough, the summarization engine that this process served often selected the non-trimmed variants of sentences, in some cases quite correctly since the trimmed version lost important content, either due to parser error or overtrimming.

Final Thoughts

Three insights guided the work presented here. (1) Although resolving some instances of VP ellipsis requires sophisticated semantic and pragmatic reasoning, not all cases are so difficult. (2) The "difficult/simple" judgment can be operationalized by exploiting linguistic principles that can be implemented within the current state of the art. (3) Many complex contexts can be automatically simplified, with the simplified versions being treatable by our ellipsis resolution methods.

The decision to permit the system to select which examples to treat and which to leave untreated is not typical in current NLP. Systems that treat overt referring expressions more typically function in one of two different modes: either they orient around an annotated corpus, which indicates which entities must be treated (the so-called "markables"); or they attempt to treat all instances of a given string. Our interest in permitting the system to select which contexts to treat derives from the agent-building paradigm. Given an input, the agent must decide if it understands it sufficiently to proceed to decision-making and action. Endowing agents with estimates of language processing confidence will, we believe, contribute to making them better collaborators with humans in the near future.

As a contribution to cognitive science, this approach to agent modeling operationalizes the notion of a "simple" context – i.e., one involving a minimal cognitive load for the agent. Orienting around a psychologically-plausible foothold like this is, we believe, essential when attempting to treat difficult linguistic phenomena such as ellipsis.

Acknowledgments

This research was supported in part by Grant N00014-09-1-1029 from the U.S. Office of Naval Research. All opinions and findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

Buyko, E., Faessler, E., Wermter, J. & Hahn, U. (2011). Syntactic simplification and semantic enrichment—trimming dependency graphs for event extraction. *Computational Intelligence* 27(4).

Davies, Mark. (2008-) *The Corpus of Contemporary American English: 450 million words, 1990-present*. Available online at http://corpus.byu.edu/coca/.

Gagnon, M. & Da Sylva, L. (2005). Text summarization by sentence extraction and syntactic pruning. *Proceedings of Computational Linguistics in the North East*, Gatineau, Québec, Canada, 26 August 2005.

Graff, D., & Cieri, C. (2003). *English Gigaword*. Linguistic Data Consortium. Philadelphia.

Hardt, D. (1997). An empirical approach to VP ellipsis. *Computational Linguistics* 23(4): 525-541.

Knight, K. & Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1).

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M. & Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics* 39(4): 885-916.

Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J. & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55-60).

McShane, M. & Babkin, P. (2015). Automatic ellipsis resolution: Recovering covert information from text. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15).*

Vanderwende, L., Suzuki, H., Brockett, C. & Nenkova, A. (2007). Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*, 43(6): 1606–1618.

Zajic, D., Dorr, B., & Schwartz, R. (2004). BBN/UMD at DUC-2004: Topiary. *Proceedings of DUC-2004*.