Assisted Nominalization for Academic English Writing

John Lee[1,2], **Dariush Saberi**[1], **Marvin Lam**[3], **Jonathan Webster**[1,2]
[1] Department of Linguistics and Translation, City University of Hong Kong
[2] The Halliday Centre for Intelligent Applications of Language Studies, City University of Hong Kong
[3] Department of English, The Hong Kong Polytechnic University
jsylee@cityu.edu.hk, dsaberi2-c@my.cityu.edu.hk,
marvin.lam@polyu.edu.hk, ctjjw@cityu.edu.hk

Abstract

Nominalization is a common linguistic feature in academic writing. By expressing actions or events (verbs) as concepts or things (nouns), nominalization produces more abstract and formal text, and conveys a more objective tone. We report our progress in developing a system that offers automatic assistance for nominalization. Given an input sentence with a complex clause, it paraphrases the sentence into a simplex clause by transforming verb phrases into noun phrases. Preliminary evaluations suggest that system performance achieved high recall.

1 Introduction

University students who are non-native speakers of English often experience significant difficulties in studying content subjects in English, due in large part to their problems with academic writing (Evans and Green, 2007). The traditional focus in computer-assisted language learning and natural language processing has been the development of algorithms for correcting grammatical errors (Dahlmeier et al., 2013) and improving sentence fluency (Sakaguchi et al., 2016). The focus of this paper, in contrast, is to help students improve their writing style in Academic English.

Our long-term goal is to help students make use of the full range of options available along what M. A. K. Halliday calls the cline of metaphoricity (Halliday and Matthiessen, 2014). This cline ranges between the clausally complex, lexically simple congruent construal of experience at one end, and the clausally simple, lexically dense metaphorical re-construal at other end. Table 1 shows paraphrases of an example sentence along this cline. We envision a system that provides assistance in moving a sentence from any point on the cline to another. As a first step towards this goal, the current system focuses on paraphrasing a complex clause (e.g., "Because she didn't know the rules, she died.") into a simplex clause ("Her ignorance of the rules caused her to die.").

The rest of the paper is organized as follows. The next section summarizes previous work in automatic paraphrasing. Section 3 describes the three components in our system: syntactic parser, nominalizer, and sentence generator. Section 4 evaluates system performance, focusing on the output of the nominalizer. Section 5 concludes and discusses future work.

2 Previous work

Previous work in automatic paraphrasing can be viewed at two levels. At the word level, research in lexical substitution (McCarthy and Navigli, 2009) and the related task of lexical simplification (Specia et al., 2012) aims to replace a word or short phrase with another, while preserving the meaning of the original sentence.

At the sentence level, most previous work focused on syntactic simplification, i.e., to reduce the syntactic complexity of a sentence by splitting a complex sentence into two or more simple sentences (Siddharthan, 2002). In terms of the cline of metaphoricity, then, it transforms a "complex clause" into a "cohesive sequence" (Table 1). Typically, the system analyzes the input sentence via a parse tree, applies manually written transformation rules (Bott et al., 2012; Siddharthan and Angrosh, 2014; Saggion et al., 2015), and then performs sentence re-generation. Our system adopts a similar architecture and also takes a complex clause as input, but works in the opposite direction on the cline: it attempts to transform the complex clause into a simplex clause.

Proceedings of the Workshop on Intelligent Interactive Systems and Language Generation (2IS&NLG), pages 26–30,
Tilburg, The Netherlands, November 5 2018.

Domain	System	Example
Cohesive sequence	conjunction	She didn't know the rules. Consequently, she died.
Complex clause	parataxis	She didnt know the rules; so she died.
	hypotaxis	Because she didnt know the rules, she died.
Simplex clause	causation	Her ignorance of the rules caused her to die.
	circumstantiation	Through ignorance of the rules, she died.
	relational process	Her death was due to ignorance of the rules.
		Her ignorance of the rules caused her death.
		The cause of her death was her ignorance of the rules.
Nominal group	qualification	Her death through ignorance of the rules.

Table 1: The cline of metaphoricity, illustrated with example paraphrases of a sentence expressing a relationship of cause (Halliday and Matthiessen, 2014).

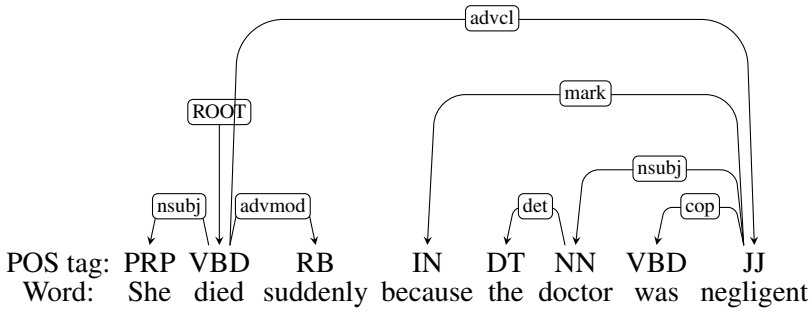
Component	Example
Input	
Syntactic Parser	Main clause = "She died suddenly" Subordinate clause = "The doctor was negligent" Linking word = "because"
Nominalizer	NP for main clause = "her sudden death" NP for subordinate clause = "the doctor's negligence"
Sentence Generator	"The doctor's negligence caused her sudden death."

Table 2: The system extracts the main and subordinate clauses of the input sentence with the *Syntactic Parser* (Section 3.1); (2) transforms the clauses into noun phrases with the *Nominalizer* (Section 3.2); and (3) links the noun phrases to produce a sentence with the *Sentence Generator* (Section 3.3).

3 Approach

Our system is a pipeline with three components (Table 2).

3.1 Syntactic parser

Given an input sentence, we use the SpaCy dependency parser (Honnibal and Johnson, 2015) to derive its syntactic tree in Universal Dependencies (Nivre et al., 2016). The system determines whether a sentence contains a complex clause by searching for the adverbial clause modifier (`advcl`) relation. If so, the system extracts the main clause from the head word of the `advcl` relation, the subordinate clause from its child word, and the linking word from the `mark` relation. In Table 2, for example, it extracts "She died suddenly" as the main clause, "the doctor was negligent" as the subordinate clause, and "because" as the linking word.

3.2 Nominalizer

Given a clause with a verb phrase, the Nominalizer matches its tree structure to the pattern shown in Table 3. It then transforms the clause into a noun phrase with the following steps:

- Identify the main verb (`verb`) and generate its nominalized form, $v2n$(`verb`). In the example in Table 3, "died" is transformed into "death". We do not treat verbs-to-be, modal

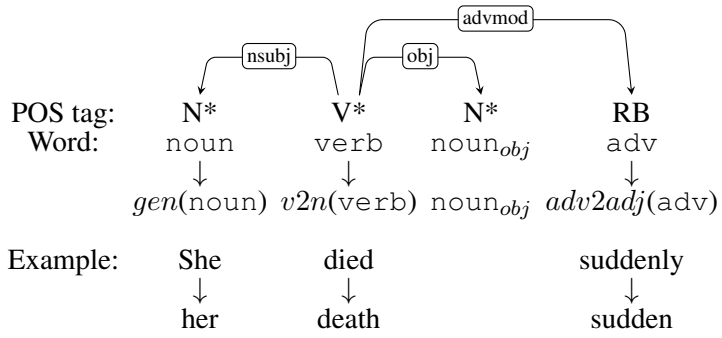
Input	
Output	*gen*(noun) *adv2adj*(adv) *v2n*(verb) of $noun_{obj}$ (Example: "her sudden death") the *adv2adj*(adv) *v2n*(verb) of noun the *adv2adj*(adv) *v2n*(verb) of $noun_{obj}$ by noun

Table 3: Nominalization rule, where $v2n$ is the mapping from a verb to a noun; $adv2adj$ is the mapping from an adverb to an adjective; and gen is the mapping from a nominative noun to its genitive form.

verbs and negated verbs, since their nominalization patterns vary considerably depending on meaning and context.

- Identify the adverb (adv), if any, and generate its adjectival form, $adv2adj$(adv). For example, "suddenly" is transformed into "sudden" in Table 3.
- Identify the direct object ($noun_{obj}$) and prepositional phrases, if any, and place them after the nominalized main verb.
- Identify the subject (noun). If the subject is a pronoun or a short noun, use the first output template in Table 3. For pronouns, gen(noun) generates its possessive form (e.g., "she" → "her"); for nouns, it appends a possessive apostrophe (e.g., "doctor" → "doctor's"). For longer noun phrases, the system prepends "of" when using the second template, or "by" when using the third template (e.g., "the doctor in the clinic" → "of/by the doctor in the clinic").

A similar rule is defined for clauses with an adjectival phrase (e.g., "The doctor was negligent"). Using an adjective-to-noun mapping $adj2n$, it rewrites the adjective into a noun (e.g., "the doctor's negligence"). Both rules operate on the following part-of-speech conversion lists:

Verb-to-noun ($v2n$) We constructed our verb-to-noun list based on the NOM entries[1] from NOMLEX (Meyers et al., 1998). For verbs not covered by NOMLEX, we retrieved their nouns in CATVAR (Habash and Dorr, 2003). When a verb is mapped to multiple nouns, we choose the one with the highest unigram frequency count in the *Google Web 1T Corpus* (Brants and Franz, 2006) that ends with a typical noun suffix[2]. This procedure yielded 7,879 one-to-one verb-noun mappings.

Adjective-to-noun ($adj2n$) We constructed our adjective-to-noun list with a similar procedure based on the NOMADJ entries from NOMLEX and verb-adjective pairs in CATVAR. There are 11,369 unique one-to-one adjective-noun mappings.

Adverb-to-adjective ($adv2adj$) We constructed our adverb-to-adjective mapping with CATVAR, with a total of 2,834 such mappings.

The current system assumes one-to-one mappings for the above, though it is clear that polysemy necessitates one-to-many mappings. For example, the verb "descend" should be nominalized as "descendance" in the context of "descendance from royalty", but as "descent" in the context of "descent from the mountain". In future work, we plan to incorporate automatic semantic disambiguation to make this distinction.

[1] We excluded the NOMLIKE entries, and those whose NOM-TYPE is SUBJECT, e.g., "teacher" for the verb "teach".

[2] '-age', '-ance', '-ce', '-cy', '-dge', '-dom', '-ence', '-ery', '-ess', '-esse', '-hood', '-ice', '-ics', '-ion', '-ise', '-ism', '-ity', '-ment', '-ry', '-ship', '-th', '-tude', '-ty', '-ure'.

3.3 Sentence Generator

The Sentence Generator takes as input the noun phrases produced by the Nominalizer for the main clause and subordinate clause. It then recombines them into a complete sentence based on the semantic relation between them. Since implicit discourse identification remains a challenging task (Braud and Denis, 2014), the system determines the relation by keyword spotting; the keywords "because", "since", "so", and "therefore" for the causal relation; "although", "despite", "even though" for the concession relation, "after", "before" for the temporal relation, etc. In Table 2, the system infers the relation as causal based on the keyword "because", and links the two noun phrases with the verb "to cause" ("The doctor's negligence caused her sudden death"). In other contexts, another linking word may be warranted, such as "to result in", "to lead to", "to be due/attributable to", "to be a result of", "to lie in" etc. Currently, our system lets the user choose the most appropriate word.

4 Evaluation

We performed a preliminary evaluation that focuses on the Nominalizer. We first describe our dataset and metric (Section 4.1), and then discuss the results (Section 4.2).

4.1 Data and evaluation metric

Our test data included 33 clauses present in 25 sentences from Wikipedia, covering causal, concession, and temporal relations. To produce the gold annotation, we asked a senior staff member at the English Language Centre at our university to rewrite these sentences in more nominalized forms, using a simplex clause whenever possible.

We applied our system on these sentences, and classified the system output into three categories:

- *Identical* to the gold annotation other than the position of the subject. For example, the two NPs "the existence of the company" and "the company's existence" would be considered identical;
- *Minor revision*, i.e., same choice of nominalized verb or adjective, but different choices for determiners or prepositions elsewhere in the NP. For example, the two NPs "a decrease in the number" versus "the decrease in the number" would fall into this category;
- *Major revision*, i.e., different choice of nominalized verb or adjective.

4.2 Results

While our system attempted nominalization for all 33 clauses, the human annotator nominalized only 18 of them. This suggests that in the other 15 cases, the system offered nominalizations that resulted in a less fluent sentence.

Among clauses that should be nominalized, the system achieved relatively high recall. Out of the 18 nominalizations, the system output is identical in 55.6%; requires minor revision in 16.7%; and major revision in 27.8%. Minor revisions were caused by subtle abstractions that are produced by nominalization. For example, the clause "(... even though) the years passed" was nominalized as "the passage of years" in the gold annotation, while the system did not delete the definite article. Major revisions were due sometimes to a less fluent choice of noun. For instance, "they are wrong ..." was paraphrased as "their error" in the gold annotation, while the system more mechanically generated "their wrongness". In other cases, they reflected different paraphrasing strategies. For example, the annotator nominalized "the stem is nice because ..." as "... is an attractive feature of the stem", rather than directly using a nominalized form of "nice".

5 Conclusion

We have presented a system that assists users in improving their Academic English by suggesting nominalizations. It applies transformation rules on dependency parse trees, and performs nominalization using two existing resources, NOMLEX (Meyers et al., 1998) and CATVAR (Habash and Dorr, 2003). Preliminary evaluations suggest that the system has high recall but low precision: when a clause can indeed be nominalized, the system is able to offer valid suggestions; it also provides suggestions, however, that would yield less natural sentences. As such, it is currently suitable for more advanced students who can judge the quality of these suggestions.

We plan to pursue three lines of research in future work. First, we hope to construct better verb-to-noun and adjective-to-noun mappings with automatic sense disambiguation. Second, we aim to raise precision by detecting sentences that are not amenable to nominalization. Finally, we plan to

train the sentence generator to rank its suggestions of words for linking the noun phrases.

Acknowledgments

The work was partially funded by a CityU-led Teaching and Learning Project in the 2016-19 Triennium (Title of the project: Meeting the Challenge of Teaching and Learning Language in the University: enhancing linguistic competence and performance in English and Chinese) from the University Grants Committee.

References

Stefan Bott, Horacio Saggion, and David Figueroa. 2012. A Hybrid System for Spanish Text Simplification. In *Proc.Workshop on Speech and Language Processing for Assistive Technologies*.

Thorsten Brants and Alex Franz. 2006. The google web 1t 5-gram corpus version 1.1. In *LDC2006T13*.

Chloé Braud and Pascal Denis. 2014. Combining Natural and Artificial Examples to Improve Implicit Discourse Relation Identification. In *Proc. COLING*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proc. 8th Workshop on Innovative Use of NLP for Building Educational Applications*.

Stephen Evans and Christopher Green. 2007. Why EAP is Necessary: A Survey of Hong Kong Tertiary Students. *Journal of English for Academic Purposes*, 6(1):3–17.

Nizar Habash and Bonnie Dorr. 2003. A Categorial Varation Database for English. In *Proc. NAACL*.

M. A. K. Halliday and C. M. I. M. Matthiessen. 2014. *Hallidays Introduction to Functional Grammar*. Routledge.

Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proc. EMNLP*.

Diana McCarthy and Roberto Navigli. 2009. The English Lexical Substitution Task. *Language Resources and Evaluation*, 43:139–159.

Adam Meyers, Catherine Macleod, Roman Yangarber, Ralph Grishman, Leslie Barrett, and Ruth Reeves. 1998. Using nomlex to produce nominalization patterns for information extraction. In *Proc. Computational Treatment of Nominals*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proc. Tenth International Conference on Language Resources and Evaluation (LREC)*.

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making It Simplext: Implementation and Evaluation of a Text Simplification System for Spanish. *ACM Transactions on Accessible Computing (TACCESS)*, 6(4).

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality. *Transactions of the Association for Computational Linguistics*.

Advaith Siddharthan. 2002. An Architecture for a Text Simplification System. In *Proc. Language Engineering Conference (LEC)*.

Advaith Siddharthan and M. A. Angrosh. 2014. Hybrid Text Simplification using Synchronous Dependency Grammars with Hand-written and Automatically Harvested Rules. In *Proc. EACL*.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 Task 1: English Lexical Simplification. In *Proc. First Joint Conference on Lexical and Computational Semantics (*SEM)*.