Evidence for Autosuggest for Syntactic Search

1st Author Name
Affiliation
Address
e-mail address

2nd Author Name
Affiliation
Address
e-mail address

ABSTRACT

Author Keywords
Search, Syntax, Grammatical Queries, Digital Humanities, Information Extraction

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Web search engines are quite effective at searches that are best expressed as keyword or (increasingly) natural language queries, but intuitive interfaces are still lacking for making syntactically structured queries such as *find all nouns modified by "her"*. Our goal is to build useful interfaces to aid humanities scholars search and analyze written literature; however, this group is often skeptical of digital tools, primarily because they are often difficult to use, according to a recent large survey [2]. Another survey found that 50% of linguists who wished to make very technical linguistic queries cannot program [7]. Despite this, most existing interfaces for structured querying require complex syntax that is akin to programming, thus reducing the likelihood that the target users will be willing or able to use the tool.

To address this gap, we conducted an experiment to investigate how grammatical relationships between words in English can be made more recognizable to ordinary people. Following the principle of recognition over recall, as well as the success of auto-suggest in search query interfaces, we hypothesized that examples would help people identify grammatical relationships more accurately rather than technical names.

Our results confirm that showing examples significantly improves the accuracy with which grammatical relationships are recognized. Participants identified grammatical relationships more accurately in all cases when they were shown examples of words or phrases that matched. Our findings also suggested that different types of relations benefited differently from words and phrases.

This result suggests that a query interface in which a user enters a word of interest and the system shows candidate grammatical relations augmented with examples from the text will be more successful than the baseline of simply naming the relation and showing gaps where the participating words appear.

Intuitive interfaces for syntactic queries can be useful not just for scholarly work but also for developing complex patterns for recognizing entities in text, such as medical terms [3, 6], and products and organizations [1].

RELATED WORK

Because trees are the traditional representation of a syntactic parse, some tools that allow querying of collections of syntactically parsed data focus on tree structures. For instance, the Linguist's Search Engine [**?**] uses a query-by-example strategy in which a user types in an initial sentence in English, and the system produces a graphical view of a parse tree as output, in addition to a nested LISP expression of the same tree. The user can either click on the tree or modify the LISP expression to generalize the query. Similarly, the popular Stanford Parser includes Tregex, which as the name suggests, allows for sophisticated regular expression search over syntactic tree structures, and Tsurgeon, which allows for manipulation of the trees extracted with Tregex [5]. Neither of these tools have been evaluated with usability studies. The Finite Structure Query tool for querying syntactically annotated corpora requires its queries to be stated in first order logic [4]. In the Corpus Query Language [**?**], a query is a pattern of attribute-value pairs.

Another approach (discussion of XML, Sparql goes here.)

A final simple alternative approach is to simply name the relation of interest and show blanks where the words that satisfy the relation would appear; this is the baseline design tested below.

According to Shneiderman and Plaisant [**?**], query-by-example has largely fallen out of favor as a user interface design approach. At the same time, a related technique, auto-suggest, has become a widely-used approach in search user interfaces with strong support in terms of its usability [**?**]. More here...

EXPERIMENT 1: DO EXAMPLES HELP?

Hypothesis

Our experiment's goal was to find out whether grammatical relationships could be made more recognizable

by showing examples of their usage. We tested two types of examples: a list of matching words and a list of matching phrases containing the relationship. These alternatives correspond to the explicitly visible and implicitly-inferred portions of a grammatical relation. The words are explicitly visible in the text, but the grammatical relationship is implicitly inferred from contextual information such as the part of speech of the verb, the relative ordering, and any accompanying words.

Our hypothesis was the following:

> H1. Grammatical relations can be made more recognizable by showing examples of words or phrases that match.

To test it, we presented participants with a series of identification tasks. In each task (Figure 1a), they were shown a list of sentences in which a particular grammatical relationship existed between two highlighted words. They were asked to identify which relationship it was from a list of four options. Using a between-subjects design, we tested different strategies for presenting these options. Our goal was to see whether participants to whom we showed example usages identified the relationships more accurately than those to whom we did not.

Variables

Presentation

We presented the choices in three different ways. The **baseline** presentation was a short label using linguistic terminology (Figure 1b), the **words** presentation was the short label accompanied by a list of words that matched (Figure 1c), and the **phrases** presentation was the short label accompanied by a list of phrases in which that relationship surfaced (Figure(1a). Figure 1 shows what the three conditions of this identification task looked like for the `amod(life, ___)` task.

Relation Type

English grammatical relationships have two dimensions of variability that our study design had to account for: different characteristics, and the fact that they involve words with two different functions.

First, grammatical relationships are not all the same, they vary in how familiar they are, the distance they span, and the variability of the wording with which they surface. Some relationships, such as the adjective modifier, are taught in schools, whereas others are not. Some, such as adverbial relations, are distinctive because adverbs usually end in 'ly'. Clausal complements and conjunctions can link words across whole sentences, whereas noun compounds only operate over adjacent words. Prepositional relationships used a fixed set of prepositions to link two word, but adverbial clauses can appear in almost any form.

Because of this variability, we had to test a number of different types grammatical relationships. We tested two main categories of relationships:

1. Clausal or long-distance relations:

   advcl Adverbial clause: *she **said** it while **smiling***

   xcomp Open clausal complement: *I **learned** to **sing***

   ccomp Clausal complement: *I **thought** that I **knew** it*

   rcmod Relative clause modifier: *the **cat**, which we **rescued**, slept*

2. Other relations:

   nsubj Subject of verb: ***he threw** the ball*

   dobj Object of verb: *he **threw** the **ball***

   amod Adjective modifier ***red ball***

   prep_in Preposition (in): *the **water** in the **bucket***

   prep_of Preposition (of): *the **piece** of **cheese***

   conj_and Conjunction (and) ***mind** and **body***

   advmod Adverbial modifier: *she **said** it **slowly***

   nn Noun compound: ***Mr. Brown***

Words

The second dimension of variability is that a relation links two words that have different functions. In the verb-subject relationship "***he threw***", "he" is a noun and "threw" is a verb. When presenting a participant with a list of sentences containing the relationship, we therefore have several options: we could keep the relationship the same and vary the two words that are linked, we could keep the relationship and one word the same, and vary the second, or we could keep all three the same.

We decided on the middle approach – to fix the relationship as well as one of the words, but to test each relationship 4 times, with different words in the two different roles. For example, the verb-subject relation `nsubj` was tested in the following four forms:

1. `nsubj(Ahab, ___)`: the sentences each contained 'Ahab', highlighted in yellow, as the subject of different verbs highlighted in pink.

2. `nsubj(captain, ___)`

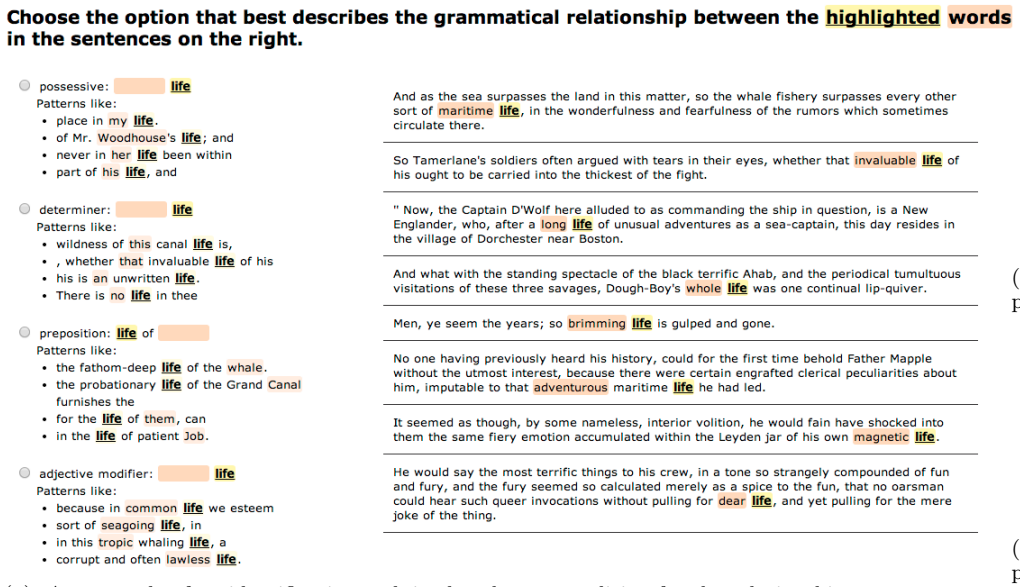
3. `nsubj(___, said)`: the sentences all contained the verb 'said', highlighted in yellow, but with different subjects, highlighted in pink.

4. `nsubj(___, stood)`

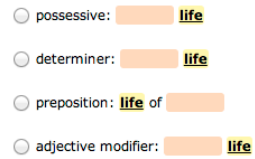
Task Variables

The tasks were all generated using the Stanford Parser on the text of *Moby Dick* by Herman Melville. When parse errors appeared, we corrected them by hand.

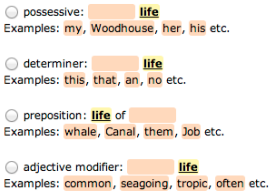
To maximize coverage, yet keep the number of tasks reasonable (around 7 or 8 minutes), we divided the relations above into 4 task sets of 3 relations each. Each relation was tested with 4 different words, making a total of 12 tasks per participant.



(a) An example of an identification task in the *phrases* condition for the relationship `amod(life, ___)` (where different adjectives modify the noun 'life'). The correct answer is 'adjective modifier' (4th option), and the remaining 3 options are distractors.



(b) The same options as they appear in the *baseline* condition



(c) The same options as they appear in the *words* condition

Figure 1: The way the choices appeared in the three experiment conditions.

The tasks were presented in the same order, and the choices were also presented in the same order: the only variation between participants was the way in which those choices were displayed. In each task, there was a 'query' word and a relationship. The participants were shown list of 8 sentences containing that relationship between the query word and other words. The query word was highlighted in yellow and the matching word in pink (Figure 1a). Their task was to identify the relationship from list of 4 choices.

To make sure that the participants could not simply guess the right answer by pattern-matching, we ensured that there was no overlap between the list of sentences shown, and the examples shown in the choices as words or phrases.

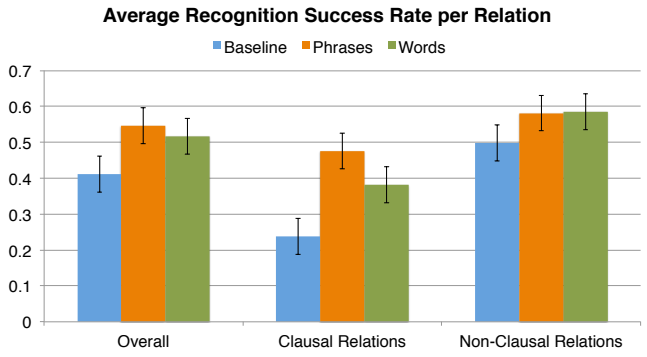


Figure 2: Recognition rates per relation under the different experiment conditions.

Participants

There were 400 participants in total, split randomly across the 4 task sets and the 3 presentations. The ability to issue grammatical search queries is relevant to many fields outside linguistics and language study. We therefore wanted to avoid having any specific backgrounds overrepresented. To achieve this, we chose Amazon's Mechanical Turk crowdsourcing platform as a source of study participants.

Participants were paid 50 cents for completing the task, with an additional 50-cent bonus if they correctly identified 10 or more of the 12 relationships. They were informed of the possibility of the bonus before starting the task.

Screening

As is difficult to ensure the quality of effort from participants from Mechanical Turk, we included a multiple-choice screening question, 'What is the third word of this sentence?" Those that answered incorrectly were eliminated.

Results

Our results (Figure **??**) confirm H1: examples improve the recognizability of grammatical relations. Participants in the **baseline** condition were significantly worse at identifying the relations than participants in conditions that showed examples (**phrases** and **words**). The average success rate (where success means that the participant correctly identified the relation) in the

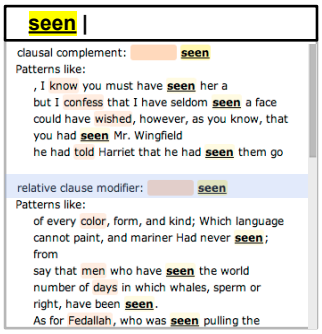


Figure 3: Mockup of auto-suggest for syntactic search on the word 'life', showing the most common grammatical relations with example phrases for each.

baseline condition was 41%, which is significantly[1] less accurate than in the two example-showing conditions: **words**: 52%, ($p = 0.00019$), and **phrases** condition : 55%, ($p = 0.00013$).

The difference between the two types of examples, **phrases** and **words**, was not significant overall, but the data revealed an interesting fact when they were compared across the different types of relations (Figure **??**). In all cases, the baseline performs worse that an example-showing presentation. However, the three different categories of relations behaved very differently with respect to whether phrases or words was better.

For the clausal relations, which operate over longer distances in sentences, the data confirmed what one might intuitively expect. Phrases, which show the usage context, significantly improved recognizability compared to the list of words or the baseline labels. The average success rate is 48% for **phrases**, which is significantly more than **words**: 38%, ($p = 0.017$), or **baseline**: 24%, ($p = 1.9 \times 10^9$).

For the other relations, there was no real difference between **phrases** and **words**, although they were both still significantly better than the baseline (words: $p = 0.0063$, phrases: $p = 0.023$).

DISCUSSION

A list of phrases is the most recognizable presentation for clausal relationships, and is as good as a list of words for the other types of relations. This implies that auto-suggest interfaces for syntactic search should use this format. A mockup of such a search box is shown in Figure 3.

Future Work

While **phrases** were slightly better than **words** overall for the non-clausal relations (Figure **??**), there was one relation for which the opposite seemed to be true. For

[1]Using the Wilcoxson signed-rank test, an alternative to the standard T-test that does not assume samples are normally distributed.

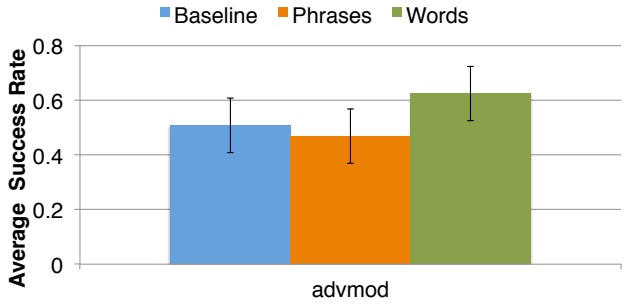


Figure 4: Average recognition success rates for the advmod relation, by presentation.

adverb modifiers (`advmod`) (Figure 4), the data seemed to suggest that **words** (0.63% success) made the relation more recognizable than **phrases** (0.47% success, $p = 0.055$), which is barely significant due to the smaller sample size (only 96 participants encountered this relation).

REFERENCES

1. Culotta, A., and McCallum, A. Reducing labeling effort for structured prediction tasks. In *AAAI* (2005), 746–751.
2. Gibbs, F., and Owens, T. Building better digital humanities tools. *DH Quarterly 6*, 2 (2012).
3. Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A. Overview of biocreative: critical assessment of information extraction for biology. *BMC bioinformatics 6*, Suppl 1 (2005), S1.
4. Kepser, S. Finite structure query: A tool for querying syntactically annotated corpora. In *EACL* (2003), 179–186.
5. Levy, R., and Andrew, G. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation* (2006), 2231–2234.
6. MacLean, D. L., and Heer, J. Identifying medical terms in patient-authored text: a crowdsourcing-based approach. *Journal of the American Medical Informatics Association* (2013).
7. Soehn, J.-P., Zinsmeister, H., and Rehm, G. Requirements of a user-friendly, general-purpose corpus query interface. *Sustainability of Language Resources and Tools for Natural Language Processing 6* (2008), 27.