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 effective for keyword searches and (increasingly) natural language queries, but intuitive interfaces are still lacking for syntactically structured queries such as *find all adjectives that modify "clothes"*. Such queries are useful in the humanities and social sciences, when scholars are attempting to characterize a concept, and also for developing complex patterns for recognizing entities in text, such as medical terms [4, 8], and products and organizations [1].

Our goal is to build interfaces to help 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 [11]. Despite this, most existing interfaces for structured querying require complex program-like syntax, 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 English words 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 than technical names.

Our results confirm that showing examples in the form of words or phrases that match significantly improves the accuracy with which grammatical relationships are recognized. Our findings also showed that different types of relations benefited differently from words and phrases.

These findings suggest 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.

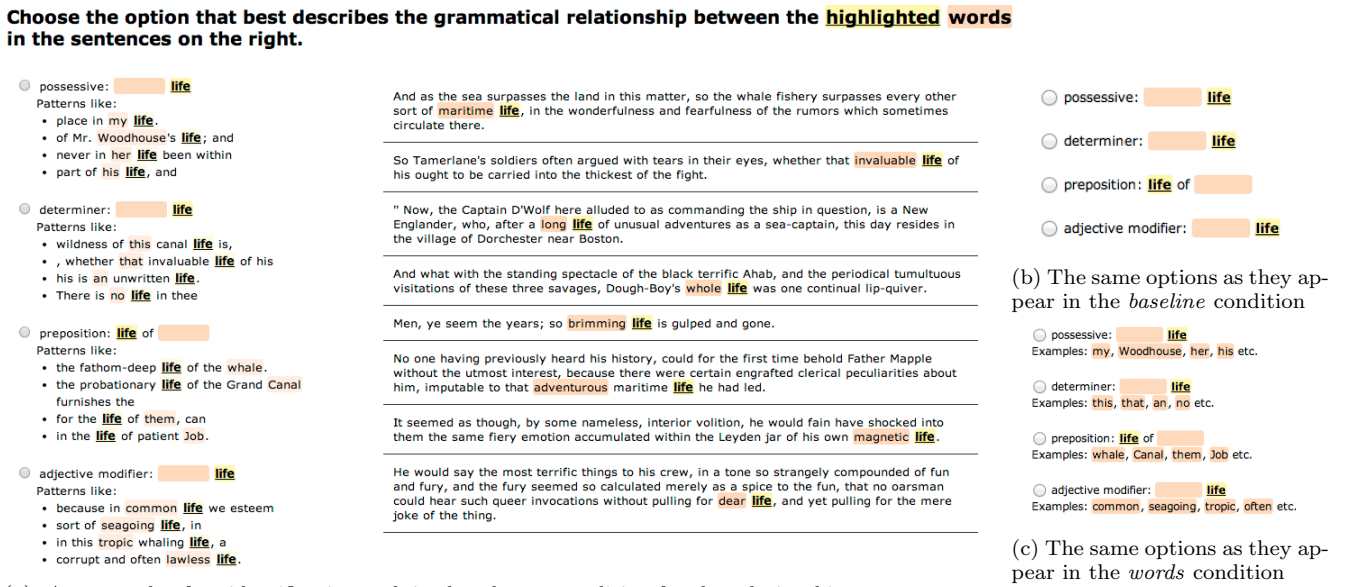
RELATED WORK
Because trees are the traditional representation of a syntactic parse, some tools that allow querying of collections of syntactically parsed data focus on tree structures. For instance, the Linguist's Search Engine [9] uses a query-by-example strategy in which a user types in an initial 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 [7]. 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 [6]. In the Corpus Query Language [5], a query is a pattern of attribute-value pairs.

Resource Description Framework (RDF) has become a popular way of storing structured data on the web. The standard query language for this type of data is SPARQL. These queries are SQL-like expressions that specify certain attribute-values and retrieve matches over other attributes by selecting, filtering, or aggregating over them.

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 [10], 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 [3]. More here...

EXPERIMENT



(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.

Our goal was to find out whether showing examples improves the recognizability of grammatical relations. We tested two types of examples: a list of matching words and a list of matching phrases. Words are explicitly visible in the text, but phrases provides contextual information that helps determine the relationship, such as the part of speech, 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 gave participants a series of identification tasks. In each task, participants were shown list of 8 sentences, each containing a particular relationship between highlighted words. They were asked to identify which relationship it was from list of 4 choices

We displayed the choices in 3 different ways (Figure 1). 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).

We used a between-subjects design. 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. We measured whether participants in the **words** or **phrases** condition identified relationships more accurately than participants in the **baseline** condition.

To avoid the possibility of participants guessing 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.

The tasks were all generated using the Stanford Parser on the text of *Moby Dick* by Herman Melville. We tested the 12 most common grammatical relationships in the novel, which fell into the two categories below.

- 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*
- 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***

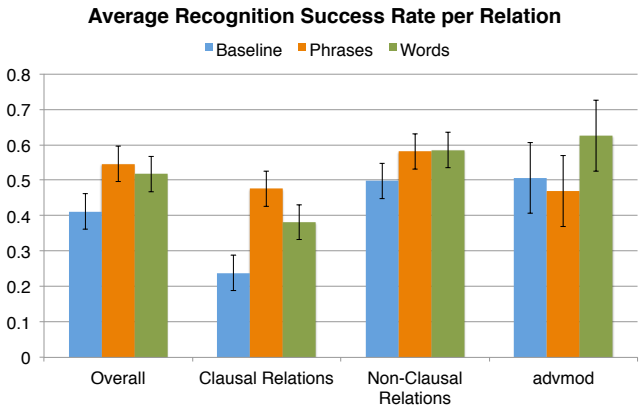


Figure 2: Recognition rates per relation under the different experiment conditions.

We tested each of the above relations 4 times, with 2 different words in each role. For example, the verb-subject relation `nsubj` was tested in the following four forms:

- `nsubj(Ahab, ___)`: the sentences each contained 'Ahab', highlighted in yellow, as the subject of different verbs highlighted in pink.
- `nsubj(captain, ___)`
- `nsubj(___, said)`: the sentences all contained the verb 'said', highlighted in yellow, but with different subjects, highlighted in pink.
- `nsubj(___, stood)`

To maximize coverage, yet keep the total task time 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.

Participants

We chose Amazon's Mechanical Turk crowdsourcing platform as a source of study participants because syntactic search is relevant to many fields outside linguistics and language study and we wanted to avoid having any specific backgrounds overrepresented. There were 400 participants, distributed randomly over the 4 task sets and the 3 presentations. Participants were paid 50c (U.S.) for completing the task, with an additional 50c 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.

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 2) confirm H1. Participants in conditions that showed examples (**phrases** and **words**) were significantly better at identifying the relations than participants in the **baseline** condition. The average success rate (where success means that the participant correctly identified the relation) in the **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%, ($p = 0.00013$).

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 non-clausal relations, there was no real difference between **phrases** and **words**, although they were both overall significantly better than the baseline (words: $p = 0.0063$, phrases: $p = 0.023$). Among these relations, adverb modifiers (`advmod`) stood out (Figure 2), because **words** (0.63% success) made the relation more recognizable than **phrases** (0.47% success, $p = 0.055$) – but the difference was barely significant, due to the smaller sample size (only 96 participants encountered this relation). This may be because the words (and not the surrounding context) is the most salient piece of information in an adverbial relation – adverbs usually end in 'ly'.

DISCUSSION

Our results imply that auto-suggest interfaces for syntactic search should show candidate relationships augmented with a list of phrases in which they occur. 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. A mockup of such a search box is shown in Figure 3.

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. Hearst, M. *Search user interfaces*. Cambridge University Press, 2009.
4. 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.
5. Jakubicek, M., Kilgarriff, A., McCarthy, D., and Rychlỳ, P. Fast syntactic searching in very large

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

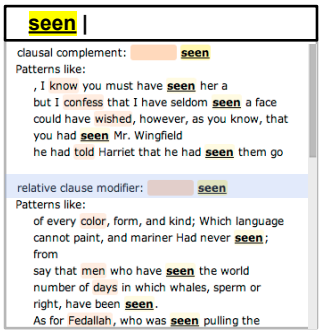


Figure 3: Mockup of auto-suggest for syntactic search on the word 'life', showing the most common grammatical relations with example phrases for each.

corpora for many languages. In *PACLIC*, vol. 24 (2010), 741–747.

6. Kepser, S. Finite structure query: A tool for querying syntactically annotated corpora. In *EACL* (2003), 179–186.
7. 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.
8. 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).
9. Resnik, P., Elkiss, A., Lau, E., and Taylor, H. The web in theoretical linguistics research: Two case studies using the linguists search engine. In *Proceedings of the 31st Meeting of the Berkeley Linguistics Society* (2005), 265–276.
10. Shneiderman, B., and Plaisant, C. *Designing The User Interface: Strategies for Effective Human-Computer Interaction, 5/e (Fifth Edition)*. Addison Wesley, 2010.
11. 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.