Improving Cloze Test Performance of Language Learners Using Web N-Grams

Martin Potthast   Matthias Hagen   Anna Beyer   Benno Stein
Bauhaus-Universität Weimar, Germany
`<first name>.<last name>@uni-weimar.de`

Abstract

We study the effectiveness of search engines for common usage, a new category of search engines that exploit n-gram frequencies on the web to measure the commonness of a formulation, and that allow their users to submit wildcard queries about formulation uncertainties often encountered in the process of writing. These search engines help to resolve questions on common prepositions following verbs, common synonyms in given contexts, and word order difficulties, to name only a few. Until now, however, it has never been shown that search engines for common usage have a positive impact on writing performance.

Our contribution is a large-scale user study with 121 participants using the Netspeak search engine to shed light on this issue for the first time. Via carefully designed cloze tests we show that second language learners who have access to a search engine for common usage significantly and effectively improve their test performance as opposed to not using them.

1 Introduction

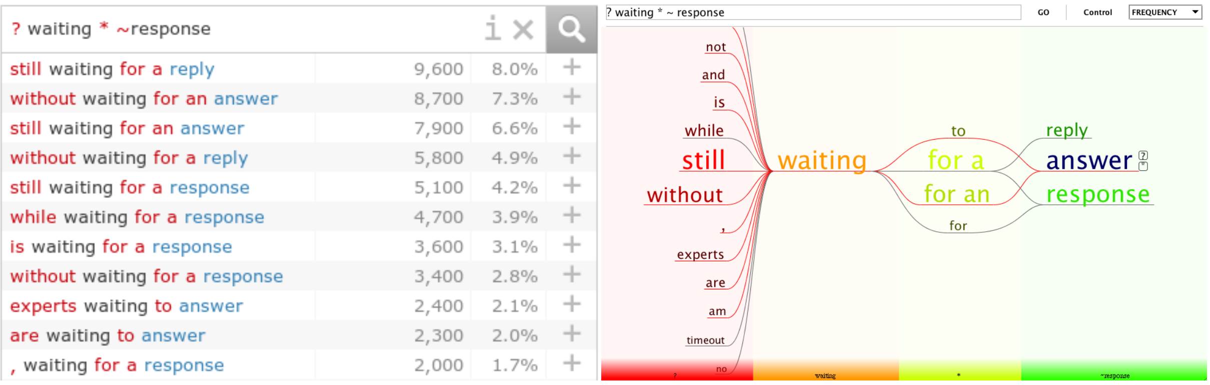
When writing texts in a second language, uncertainties on specific formulations regularly come up. Even experienced second language writers may sometimes be in doubt about the preposition following a verb or what word order to choose. In this paper, we study search engines for common usage (usage search engines, for short) that aim at assisting second language writers to cope with their uncertainties. These search engines allow for phrasal queries that include wildcards at positions where a user is not sure what to write. The search results typically consist of a list of phrases matching the query's expression—the wildcards filled with formulations. The returned phrases are ranked by their commonness of being used in everyday writing, where a phrase's commonness is estimated by its occurrence frequency in a collection of web n-grams. The occurrence frequencies are usually not hidden from the user but displayed alongside each phrase, either implicitly or explicitly. This way, the users of usage search engines have a way of judging whether a phrase is commonly used by others. Figure 1 (left) shows an example search result.

Target audience of usage search engines is language learners who have mastered basic vocabulary and grammar but whose language proficiency in terms of their feeling for language usage is still worse than that of a native speaker. Until recently, there has been hardly any technological support for them, so they could only resort to studying abstract style guides, consuming foreign language media, and language study travels in order to improve their usage skills. Today, three public usage search engines are available. The first one, called Netspeak (Stein et al., 2010), is developed at our group since 2008. It was followed by PhraseUp and Linggle (Boisson et al., 2013), which have been released in 2011 and 2013.[1] Moreover, there is Google's N-Gram Viewer prototype (Michel et al., 2011), which has a different purpose and target audience but visualizes n-gram usage over time.

All of these search engines provide a way to quantify the commonness of a phrase and thus have the potential to become important tools for second language learners. That is, if they work as advertised. Until now, it has not at all been clear whether writers can actually benefit from the information distilled from analyzing n-gram occurrence frequencies, or whether they are easily misled, for example, by noisy

[1] Netspeak is freely available at `www.netspeak.org`, PhraseUp at `www.phraseup.com`, and Linggle at `www.linggle.com`.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 962–973, Dublin, Ireland, August 23-29 2014.



? waiting * ~response

still waiting for a reply	9,600	8.0%
without waiting for an answer	8,700	7.3%
still waiting for an answer	7,900	6.6%
without waiting for a reply	5,800	4.9%
still waiting for a response	5,100	4.2%
while waiting for a response	4,700	3.9%
is waiting for a response	3,600	3.1%
without waiting for a response	3,400	2.8%
experts waiting to answer	2,400	2.1%
are waiting to answer	2,300	2.0%
, waiting for a response	2,000	1.7%

Figure 1: Netspeak's two alternative interfaces: search results can either be displayed as textual ranked list of phrases alongside frequencies (left), or as WordGraph visualization (right) (Riehmann et al., 2011), where the frequencies determine various aspects of the visualization. The WordGraph is particularly suited to handling multiple wildcards per query. The participants of our user study used primarily the textual interface, since they did not require more than one or two wildcards for solving the cloze tests.

data. Our contribution is to shed light on this issue for the first time and to conduct a large-scale user study with 121 language learners aged 14–18, measuring their performance when using our Netspeak search engine to solve cloze tests. The study ascertains the positive impact of Netspeak and by extension, usage search engines in general; moreover, it shows the low barrier to entry of Netspeak's user interface.

The paper is organized as follows: after a detailed discussion of related work in Section 2, Netspeak's retrieval engine is formally described in Section 3 as background for the design of our user study and as an example of how such search engines work internally. Section 4 reports on our user study and provides a statistical analysis of our findings. The paper closes with a conclusion and an outlook into future work.

2 Related Work

Carrying out research and development on usage search engines is an interdisciplinary effort that requires expertise from information retrieval, information visualization and interface design, as well as domain knowledge from computer linguistics. Therefore, we divide our review of related work into four parts: (1) existing search engines and web services, (2) retrieval engines and wildcard search from the perspective of information retrieval, (3) search result visualization, and, (4) writing support systems dedicated to second language writers.

2.1 Public Search Engines and Web Services

There are currently three public search engines and one public prototype that fall into the category of search engines for common usage, namely Netspeak (Stein et al., 2010), PhraseUp, Linggle (Boisson et al., 2013), and the Google N-Gram Viewer (Michel et al., 2011). All of them index large n-gram corpora, and their search interfaces are primarily dedicated to returning results that allow their users to judge the commonness of a phrase compared to alternative phrases. We distinguish the former three search engines from the latter mainly by its target audience. While the former target average web users, the latter targets professional linguists and humanities researchers. To the best of our knowledge, our paper is the first to investigate the effectiveness of such search engines for the use case of assisting writers, thereby underpinning these efforts.

Moreover, a number of other linguistic search engines are available, such as WebCorp Live (Kehoe and Renouf, 2002), WebAsCorpus (Fletcher, 2007), and the Linguist's Search Engine (Resnik and Elkiss, 2005). These search engines cannot be readily used for usage search as defined above, since they work more like concordancers in that they only retrieve usage examples and present them in context, disregarding usage commonness. Again, their target audience is professional linguists rather than laymen users, let alone second language learners. While they may still be applied in the context of language learning, the search interfaces of these search engines are not sufficiently tailored to this domain.

Another category of related web services that are readily available to second language learners include style and grammar checkers, such as Grammarly, PaperRater, SlickWrite, AfterTheDeadline (Mudge, 2010), the Hemingway App, GrammarBase, etc. From what can be said by analyzing their features, all of these services are based on a collection of basic style and grammar rules that can be checked automatically with some degree of confidence in their recommendations. However, none of the services we found make any recommendations with regard to usage commonness, i.e., they do not identify uncommon formulations or make recommendations for more common ones.

2.2 Information Retrieval Models and Indexes for Wildcard Search

The retrieval models employed by usage search engines are hardly ever discussed in the literature cited above. One of the few exceptions is Netspeak (Stein et al., 2010), where the retrieval model has been a contribution in itself since it is tailored specifically to its application domain. For the lack of discussion of the finer details of how the above search engines work, it can be assumed that they do not employ a specifically tailored retrieval approach. Nevertheless, when reviewing the information retrieval literature for retrieval models that support linguistic queries or wildcard queries, a number of sources can be found.

Cafarella et al. (2005, 2007) study indexing methods that are particularly suited to support queries comprising parts-of-speech as wildcards. They introduce so-called neighborhood indexes whose disk accesses required to answer a query are on the order of the number of non-wildcard terms in a query. Rafiei and Li (2009) develop a wildcard search engine that supports linguistically rich wildcards in order to support information extraction from the web, which employs a preprocessor for queries, and a postprocessor for search results on top of a traditional web search engine. The approach does not create a tailored index but translates the wildcard queries into flat queries that can be answered by traditional search engines. Sekine (2008) explores the trie data structure as an alternative to inverted indexes when indexing large-scale n-gram corpora. The approach is limited to short n-grams ($n < 10$) to be feasible, which can be a strong point in terms of retrieval speed. Netspeak's retrieval engine is also intentionally restricted to small values of n, but uses minimal perfect hash functions instead of tries to maximize retrieval performance.

While all of the aforementioned approaches support shallow linguistic wildcards, or only basic wildcards, Tsang and Chawla (2011) propose a method to support regular expressions. Doing so involves various trade-offs between retrieval performance and index size. Further, a search engine like this may be only useful to experts, but not second language learners. Again, all of the aforementioned contributions target either professional linguists or they are meant to facilitate automatic usage, instead of supporting average writers.

2.3 Visualization of Usage Search Results

An important part of every search engine is its user interface. Since usage search engines are still in their infancy, their user interfaces have not been studied in-depth, so far. As a first attempt to close this gap, we developed and analyzed two alternative user interfaces for Netspeak in a previous work, one textual interface and one using a tailored visualization that was specifically developed for usage search engines, the so-called WordGraph (Riehmann et al., 2011). Figure 1 shows them side-by-side. The textual interface displays search results in the form of a tabular list, where each row lists an n-gram matching the wildcard query alongside its absolute and relative occurrence frequency. If a query comprises more than one wildcard, situations arise where this linear ranking of n-grams is insufficient to grasp the true distribution of formulations that may be used instead of the wildcards. The WordGraph therefore visualizes the search results as a horizontal graph, so that the i-th word of an n-gram is displayed as a node on the i-th level of the graph. Paths from left to right through the graph correspond to n-grams found in the result set returned by Netspeak. A user study that investigated the fitness of the WordGraph to serve as a user interface for specific search tasks found that study participants prefer the WordGraph over the textual user interface when the number of wildcards increases (Riehmann et al., 2012). The user study we report on in this paper is based solely on the textual user interface, since most of our cloze tests can be solved by using one wildcard.

2.4 Writing Support for Second Language Learners

"*For writers of English as a Second Language (ESL), useful editorial assistance geared to their needs is surprisingly hard to come by*," and "[...] *there has been remarkably little progress in this area over the last decade*," observe Brockett et al. (2006) about the state of the art. This is despite the fact that English is the second language of most people who speak English today.[2] A recent overview of technology to detect grammatical errors of language learners is given by Leacock et al. (2010), whereas computer feedback for second language learners is mostly studied within pedagogical research under the label of computer-aided language learning (CALL). There, classroom systems are being deployed on a small scale to measure their effects on student learning performance. The development of usage search engines in general, our Netspeak engine in particular, and the user study contributed in this paper may be considered first steps toward the development of new, better technologies that specifically target the needs of second language learners and writers.

3 Netspeak: A Search Engine for Common Usage Based on Web N-Grams

As a background for our user study and as an example of how usage search engines work internally, this section briefly describes Netspeak and its retrieval engine.[3] The main building block of Netspeak is a query processor tailored to the following task: given a wildcard query q and a set D of n-grams, retrieve those n-grams $D_q \subseteq D$ that match the pattern defined by q. To solve this task, we have developed an index-based wildcard query processor addressing the three steps indexing, retrieval, and filtering, as illustrated in Figure 2 (middle).

3.1 Query Language

Netspeak utilizes a query language defined by the EBNF grammar shown in Figure 2 (left). A query is a sequence of literal words and wildcard operators, wherein the literal words must occur in the expression sought after, while the wildcard operators allow to specify uncertainties. Currently five operators are supported:

- the question mark (`?`), which matches exactly one word;
- the asterisk (`*`), which matches any sequence of words;
- the tilde sign in front of a word ($\sim$`<word>`), which matches any of the word's synonyms;
- the multiset operator (`{<words>}`), which matches any ordering of the enumerated words; and,
- the optionset operator (`[<words>]`), which matches any one word from a list of options.

The textual interface displays the search results for the given query as a ranked list of phrases, ordered by decreasing absolute and relative occurrence frequencies. This way, the user can find confidence in choosing a particular phrase by judging both its absolute and relative frequencies. For example, a phrase may have a low relative frequency but a high absolute frequency, or vice versa, which in both cases indicates that the phrase is not the worst of all choices. Furthermore, the textual web interface offers example sentences for each phrase, which are retrieved on demand when clicking on a plus sign next to a phrase. This allows users who are still in doubt to get an idea of the larger context of a phrase.

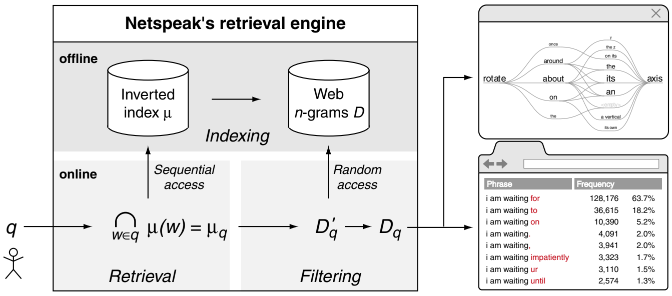
3.2 Retrieval Engine

The indexing step is done offline. Let V denote the set of all words found in the n-grams D, and let $\hat{D}$ denote the set of integer references to the storage positions of the n-grams in D on hard disk. During indexing, an inverted index $\mu : V \rightarrow \mathcal{P}(\hat{D})$ is built that maps each word $w \in V$ to a sorted list $\mu(w) \subseteq \hat{D}$, where $\mu(w)$ is comprised of exactly all references to the n-grams in D that contain w.

[2] `http://en.wikipedia.org/wiki/English_language#Geographical_distribution`

[3] Extended versions of this section can be found in previous publications on Netspeak's WordGraph visualization (Riehmann et al., 2011; Riehmann et al., 2012).

EBNF grammar of Netspeak's query language		
query	=	{ word | wildcard }$_1^5$
word	=	([apostrophe] (letter { alpha })) | " , "
letter	=	" a " | ... | " z " | " A " | ... | " Z "
alpha	=	letter | " 0 " | ... | " 9 "
apostrophe	=	" ' "
wildcard	=	" ? " | " * " | synonyms | multiset | optionset
synonyms	=	" ~ " word
multiset	=	" { " word { word } " } "
optionset	=	" [" word { word } "] "



Phrase	Frequency	
i am waiting for	128,176	63.7%
i am waiting to	36,615	18.2%
i am waiting on	10,390	5.2%
i am waiting.	4,091	2.0%
i am waiting,	3,941	2.0%
i am waiting impatiently	3,323	1.7%
i am waiting ur	3,110	1.5%
i am waiting until	2,574	1.3%

Figure 2: Netspeak at a glance (Riehmann et al., 2012): the left table shows Netspeak's query language as an EBNF grammar, the middle figure overviews its retrieval engine, and the right figure shows an example of search results as shown to its users. Given a query q, the intersection of relevant postlists yields a tentative postlist μ_q, which then is filtered and presented as a ranked list. The index μ exploits essential characteristics that are known a-priori about possible queries and the n-gram set D.

The list $\mu(w)$ is referred to as posting list or postlist. Since D is invariant, μ can be implemented as an external hash table with $O(1)$-access to $\mu(w)$. For μ being space-optimal, a minimal perfect hash function based on the CHD algorithm is employed (Belazzougui et al., 2009).

The two online steps, retrieval and filtering, are taken successively when answering a query q. Within the retrieval step, a tentative postlist $\mu_q = \bigcap_{w \in q} \mu(w)$ is constructed; μ_q is the complete set of references to n-grams in D that contain all words in q. The computation of μ_q is done in increasing order of postlist length, whereas each $\mu(w)$ is read sequentially from hard disk. Within the filtering step, a pattern matcher is compiled from q, and D_q is constructed as the set of those n-grams referenced in μ_q that are accepted by the pattern matcher. Constructing D_q requires random hard disk access. Basically, this approach corresponds to how web search engines retrieve documents for a given keyword query before ranking them. In what follows, we briefly outline how the search in D is significantly narrowed down.

With an inverted index that also stores specific n-gram information along with the keywords, the filtering of μ_q can be avoided. In this regard, we distinguish the queries that can be formulated with Netspeak's query language into two classes: fixed-length queries and variable-length queries. A fixed-length query contains only wildcard operators that represent an a-priori known number of words, while a variable-length query contains at least one wildcard operator that expands to a variable number of words. For example, the query `fine ? me` is a fixed-length query since only 3-grams in D match this pattern, while the query `fine * me` is a variable-length query since n-grams of length $2, 3, 4, \ldots$ match. Obviously, fixed-length queries can be answered with less filtering effort than variable-length queries: simply checking an n-gram's length suffices to discard many non-matching queries. The query processor first reformulates a variable-length query into a set of fixed-length queries, which then are processed in parallel, merging the results.

Moreover, the retrieval engine employs pruning strategies so that only relevant parts of a postlist are read during retrieval, presuming sorted postlists. Head pruning means to start reading a postlist at some entry within, without compromising recall. Given a query q, let τ denote an upper bound for the frequencies of the n-grams in q's result set D_q, i.e., $d \in D_q$ implies $f(d) \leq \tau$. Obviously, in all postlists that are involved within the construction of D_q, all entries whose n-gram frequencies are above τ can safely be skipped, whereas τ is determined in a preprocessing step as the lowest occurrence frequency of a sub-sequence of q that does not include wildcards. Up to this point, the retrieval of n-grams matching a query q is exact—but, not all n-grams that match a query are of equal importance. We consider this fact by applying tail pruning for postlists that are too long to be read at once into main memory. As a consequence, less frequent n-grams that might match a given query can be missed.

3.3 The Web n-Gram Collection

To provide relevant suggestions, a wide cross-section of written text on the web is required. Currently, Netspeak indexes the Google n-gram corpus "Web 1T 5-gram Version 1" (Brants and Franz, 2006), which consists of 42 GB of phrases up to a length of $n = 5$ words along with their occurrence frequencies on the web in 2006. This corpus has been compiled from approximately 1 trillion words extracted from the English portion of the web, totaling more than 3 billion n-grams. Two post-processing steps were applied: case reduction and vocabulary filtering. For the latter, a white list vocabulary V was compiled and only n-grams whose words appear in V were retained. The vocabulary V consists of the words found in the Wiktionary and various other dictionaries, complemented by words from the 1-gram portion of the Google corpus whose occurrence frequency exceeds 10 000. After post-processing, the size of the corpus has been reduced by about 54%.

3.4 Retrieval Performance in Practice and Public Availability

In practice, the described techniques enable Netspeak to provide search results at a speed similar to modern web search engines. Results are usually returned within a couple of milliseconds. Whenever a user stops typing for more than 300 milliseconds, the current input is submitted as an "instant" query without need for a click. That way, the "search experience" with Netspeak is similar to what users expect from web search engines.

Netspeak is freely available online and has about 300 distinct users on a working day who submit about 2500 queries (half the workload on weekends). Most of its users are returning users. From their feedback and from our own experience, we know that Netspeak helps to resolve uncertainties on formulations in the daily process of writing papers, proposals, etc. However, in the following section we attempt to capture Netspeak's effectiveness in a controlled user study.

4 User Study on the Effectiveness of Usage Search Engines

It is generally assumed that usage search engines are useful, say, that they provide valuable feedback that leads to improved writing. To empirically confirm this "usefulness" assumption, we conduct systematic tests with experienced language learners and analyze whether a usage search engine enables them to improve their writing. We choose Netspeak as a representative of usage search engines for our study.

Our study's underlying rationale is to model the use case of usage search engines by solving cloze tests. In a cloze test, a word or a phrase is removed from a sentence and the participant has to replace the missing words. Although we followed standard procedures on constructing cloze tests (Sachs et al., 1997), it should be noted that our usage of cloze tests is not as originally intended (Taylor, 1953). We do not assess a language learner's reading skills, but use the cloze test to model word choice, which resembles the use case of usage search engines very well. For each participant, we provide two different cloze test questionnaires. The first has to be solved without any help, whereas for the second, participants are allowed to use the search engine. Besides evaluating the answers, we also analyze the submitted search queries.

4.1 Experiment 1: General Usage, Average Learners

In the first experiment, we examine whether the search engine in general can support users in resolving uncertainties on formulations modeled by cloze tests. Our hypothesis is that using a usage search engine helps to improve a human's performance in such tests.

Experimental Design To test our hypothesis, we conduct an empirical study with a within-subjects design (Lazar et al., 2010). This means that our participants are exposed to a cloze test without the help of a search engine and then to another cloze test where our chosen usage search engine is allowed.

The to-be-solved cloze tests are carefully constructed under the guidance of a university-level English teacher who is a native English speaker. From several language learner textbooks, we selected questions in order to have an equal mix of two easy, four medium, and three hard questions for two different cloze test questionnaires A and B (see Appendix A and B).

In order to have objectively comparable test cases, the English teacher provided four possible answers for each of the nine questions from test A and B, from which participants had to choose one in each case. This way, the participants do not have to rely on their subjective own vocabulary knowledge.

Table 1: Results of our user study on the impact of usage search engines on language learners.

Experiment	**Question difficulty**	**Questions answered**										
		manually				with search engine available						
						but not used			and used			
		✓	×	–	**sum**	✓	×	–	✓	×	–	**sum**
Average Learners	easy	17	41	0	**58**	7	2	1	42	6	0	**58**
	medium	61	100	3	**164**	25	16	1	88	34	0	**164**
	hard	37	72	2	**111**	4	22	2	18	62	1	**111**
	all	115	213	5	**333**	36	40	4	149	102	1	**333**
Highly Experienced Learners	easy	11	5	0	**16**	10	1	0	4	1	0	**16**
	medium	27	17	0	**44**	24	2	0	14	3	1	**44**
	hard	18	12	0	**30**	8	8	0	4	10	0	**30**
	all	56	34	0	**90**	42	11	0	22	14	1	**90**
Specific Operators	easy	147	29	1	**177**	28	2	1	135	11	0	**177**
	medium	117	57	3	**177**	20	6	1	123	24	3	**177**
	hard	135	40	2	**177**	31	5	2	130	18	1	**177**
	all	399	126	6	**531**	79	13	4	378	53	4	**531**

Search engine not used **Search engine used**

Experiment	**Search engine used vs not used**		
	p-value	effect size	
Average Learners	**0.0000**	0.73	large
Highly Exp. Learners	0.7030	0.12	small
Specific Operators	**0.0000**	0.58	large

In the left table, ✓ denotes correct answers, × denotes wrong answers, and – denotes unanswered questions.

To evaluate the statistical significance and the effect size, we distinguished cloze test answers for the conditions "Search engine not used" and "Search engine used" in the left table.

The brackets below the bottom row of the left table indicate which cases fall under what condition.

The English teacher first chose the questions independent of knowing the indexed n-grams of the search engine. In a "postprocessing" step, the chosen answers for the questions are checked for existence in the n-gram vocabulary of the search engine. This always was the case, although sometimes the queries required to retrieve them were different from the exact context around the cloze test's missing word. This check ensured that there was a chance of answering each individual question in the cloze tests with the search engine.

During the experiment, the use or non-use of the search engine is the independent variable. The dependent variable is the number of correct answers per questionnaire. There also are confounding variables like whether our engine really was used when it was allowed, the time needed to type queries, or the different numbers of answered questions with and without the search engine. We will further elaborate on how we deal with these variables in the following description of the experimental process.

Experimental Process From three different local high schools, 43 German pupils (23 female, 20 male; mean age 16.2, $SD = 1.2$) with five or more years of English courses participated in six groups. None of the participants had any previous experience with any usage search engine.

When a group arrived in our lab, they were randomly assigned to a lab seat; questionnaire A or B were distributed ensuring that neighboring participants had a different question set. This way, the test distribution was random and the participants could not collaborate (which was also ensured by their accompanying "watchdog" teachers). After seven minutes, the first questionnaires were collected and a short five minute introduction to the search engine and its operator set was given. To ensure that the pupils really followed the introduction, we provided the chance of winning small prices based on correctly answering a question on the underlying technique of usage search engines—the index—in an exit questionnaire. After that, each participant had to solve the opposite questionnaire (A when the first was B, and vice versa) but was allowed to use the search engine this time. In pilot studies, we noticed that pupils of that age often need a lot of time for typing their search queries on a standard keyboard. Thus, we allowed 10 minutes for the second questionnaire. This confounding variable of different timing for the questionnaires could not be avoided. Otherwise, most participants would not have had the chance to complete all questions. In order to check whether our participants actually used the search engine, we logged their querying behavior and manually identified the questions which they had answered without using the search engine.

Results and Discussion Since not all participants answered all questions for both cloze tests, we excluded the six participants from the following analyses, who had a difference of more than one between the number of answered questions for either test.

The aggregated numbers on questionnaire performance for the remaining 37 individuals are given in the first block of rows of Table 1 ("Average Learners"). Note that the ratio of correct vs. incorrect answers goes up when the search engine was used: on average, an individual answered two more questions correctly. Especially interesting is that the short five minute introduction was sufficient for that effect which shows the strength of the textual interface. To statistically estimate the per-individual effect, we compare the ratio of correct answers among all answers when the search engine was used to the ratio when it was not used (note that this includes the questions where the engine was allowed but was not used; i.e., columns “manually” and “but not used” in Table 1). According to the Shapiro-Wilk test (Razali and Wah, 2011), the individual participants’ ratios are not normally distributed for either condition (engine used vs. not used) such that we choose a non-parametric significance test (Lazar et al., 2010). For our within-subjects design with ratio data and two to-be-compared samples, the Wilcoxon signed rank test is known as a suitable significance test (Lazar et al., 2010). For the 37 participants’ ratios we get a p-value below 0.001 and thus can reject the null hypothesis that the ratios’ distributions are equal. Further estimating the effect size for the Wilcoxon signed rank test, we obtain a value of 0.73 which corresponds to a large effect (Cohen, 1988; Fritz et al., 2012). This result supports our prediction that the search engine can help resolve writing uncertainties.

We also studied the query logs of our participants. Per cloze test question, they submitted 4–5 queries with 2–3 terms on average (a wildcard is counted as a term). The last query in each such “search session” for a single question typically was 3–4 terms long. Almost all participants only used the `?`-operator and most participants chose the strategy of querying with context before and after the operator. Having only context before or only after the operator are less successful strategies with higher error ratios.

4.2 Experiment 2: General Usage, Highly Experienced Learners

In our neighborhood, there also is an international high school, where German pupils have all their classes taught in English. Obviously, such pupils have a much higher experience speaking and writing English than our participants from Experiment 1. For a second experiment, we invited pupils from the international school to our lab. Our hypothesis is that the pupils from the international school will have to use the search engine less frequently but still can benefit from it for individual questions.

Experimental Design and Process We used the same questionnaires, time constraints, and logging strategies as in Experiment 1. From the international school, 12 German pupils (7 female, 5 male; mean age 16.5, $SD = 0.7$) participated in two groups. These pupils are taught all their courses in English for five and more years. None of them had any previous experience with usage search engines. The experimental process was as in Experiment 1.

Results and Discussion Again, not all participants answered all questions for both cloze tests; we excluded the two participants from the following analyses, who had a difference of more than one between the number of answered questions for either test.

The aggregated numbers on questionnaire performance for the remaining 10 individuals are given in the second block of rows of Table 1 (“Highly Experienced Learners”). As expected, the highly experienced pupils used the search engine very rarely. This is not too surprising since our questionnaires were designed with an average German pupil in mind; many questions seemed too easy to the internationals which they also indicated in their exit questionnaires. Still, on a per-question basis, for the medium and difficult questions where the pupils used the search engine, they slightly improved their performance. However, the sample and the effect size are too small to draw any reliable conclusions.

The experiment shows that the highly experienced pupils indeed did not use our engine often. However, the predicted benefit for them cannot be confirmed from our small sample. It is thus an interesting open task to conduct a larger study with highly experienced users and more difficult questions.

4.3 Experiment 3: Specific Operators, Average Learners

Our first experiment revealed that most participants used the `?`-operator to solve the tasks. We thus designed a third experiment specifically targeted at the options, synonyms, and word-order operators of our Netspeak search engine. Our hypothesis is that each individual operator helps improve a human’s performance in cloze tests targeted at the individual operator.

Experimental Design As in Experiment 1, we asked the university-level English teacher to design two cloze test questionnaires (see Appendix C and D); for each operator with an easy, a medium, and a hard question. Here, the questions for the option operator are of a similar kind as the questions from Experiment 1. Four alternatives are given, but the participants are asked to use the option operator `[]` and not the `?`-operator. For synonyms, a complete sentence is given and for a specified word, the best among four given potential synonyms is requested. As for the word order operator, a two-word phrase is missing from the sentence and the two different word orders are provided as options. Like in Experiment 1 and 2, the explicit answer options ensure that the test is objective and not subjective. In a second development step, the questions were checked for solvability using the search engine just like in Experiment 1.

Experimental Process From three different local schools, 66 pupils (45 female, 21 male; mean age 15.9, $SD = 1.4$) participated in six groups. None of the pupils participated in Experiment 1 or 2 nor had they any previous experience with usage search engines. These pupils have learned English in their schools for at least five years. The schedule was similar to Experiment 1 with an emphasis on the three tested operators in the introductory explanations on Netspeak. In the questionnaires, the pupils were asked to use only the specific operator for the respective queries. Logging their queries, we are able to exclude solutions obtained by using a not-allowed operator.

Results and Discussion Again, not all participants answered all questions for both cloze tests; we excluded the seven participants from the following analyses, who had a difference of more than one between the number of answered questions for either test.

The aggregated numbers on questionnaire performance for the remaining 59 individuals are given in the third block of rows of Table 1 ("Specific Operators"). Note that the ratio of correct vs. incorrect answers goes up when the search engine was used: one to two more questions correctly answered on average. As in Experiment 1, the short five minute introduction is sufficient for that effect which shows the strength of our interface. To statistically estimate the per-individual effect, we compare the ratio of correct answers among all answers when the search engine was used to the ratio when it was not used (note that this includes the questions where the engine was allowed but was not used; i.e., columns "manually" and "but not used" in Table 1). For the 59 participants' ratios, we get a p-value below 0.001 and thus can reject the null hypothesis that the ratios' distributions are equal. Further, estimating the effect size for the Wilcoxon signed rank test, we obtain a value of 0.58 which corresponds to a large effect (Cohen, 1988; Fritz et al., 2012). Again, the result supports our prediction that usage search engines can help resolve writing uncertainties.

However, a deeper analysis reveals that the large effect is due to the synonym operator. Only for that operator, a statistically significant performance difference and a large effect size can be shown. For the other two operators, the null hypothesis of no performance difference cannot be rejected. This is in line with the exit questionnaire findings, where the pupils reported the synonym operator to be very helpful while the other questions were perceived as rather easy. In the query log analyses, we found that context before and after the wildcard had a similarly positive effect as before and was generally better than adding context only before the wildcard.

5 Conclusion and Future Work

Search engines for common usage have the potential to become an important tool for second language writers and learners. The possibility to check one's language against what is commonly written forms a unique opportunity to improve one's writing on-the-fly. Such information has not been available at scale so far. Our user study shows that usage search engines can indeed help second language writers solve uncertainties about formulations. Modeling writing uncertainties by carefully designed cloze tests, we are able to show a significant improvement when experienced language learners use the search engine.

Highly experienced language learners represented by our study participants from an international school, however, did not use the search engine often enough to draw meaningful conclusions. This can probably be attributed to the fact that the cloze tests were not tailored to their level of language proficiency. Therefore, the question of whether also highly experienced writers and learners, or even native speakers, can benefit from such search engines remains open and is left for future work.

Another missing piece in determining the effectiveness of usage search engines is whether their users actually learn something while using them, or whether users frequently submit the same or similar queries again and again. Our user study was not designed to answer this question, since our participants were only around for about 30 minutes for organizational reasons. Even measuring effects on short-term memory is rendered infeasible in this time frame. A longitudinal study would be ideal, in this case, but we also see an exciting, data-driven way to approach this. By analyzing the query logs of Netspeak, which is currently being used hundreds of times per day, we can track returning users. We can then study their online search behavior to determine if and how often they return to submit similar queries, which allows us to draw conclusions about their learning success. More generally, the query logs of usage search engines may form a unique opportunity to observe language learners "in the wild" as opposed to the laboratory.

Finally, regarding the user interface of usage search engines, our user study has revealed ways to improve them. For example, the interface must be optimized for faster typing (especially on mobile devices) as we observed that the pupils were not adept to entering special characters on standard keyboards, which resulted in slow typing speed. Besides this, our user study also showed that the current state of Netspeak's textual user interface as well as the simplified wildcard query language is easy enough to be understood in less than a minute by any newcomer, which demonstrates the low barrier to entry that search engines for common usage have right now.

Acknowledgements

We thank the anonymous participants of our user study as well as Tim Gollub, Martin Trenkmann, Michael Völske, Howard Atkinson, Johannes Kiesel, Matthias Busse, and Alexander Herr for their help in organizing the user study.

References

Djamal Belazzougui, Fabiano C. Botelho, and Martin Dietzfelbinger. 2009. Hash, Displace, and Compress. In *Proceedings of ESA 2009*, pages 682–693.

Joanne Boisson, Ting-Hui Kao, Jian-Cheng Wu, Tzu-Hsi Yen, and Jason S. Chang. 2013. Linggle: A Web-scale Linguistic Search Engine for Words in Context. In *Proceedings of ACL 2013 (Demos)*, pages 139–144.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1. Linguistic Data Consortium LDC2006T13.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of ACL 2006*, pages 249–256.

Michael J. Cafarella and Oren Etzioni. 2005. A Search Engine for Natural Language Applications. In *Proceedings of WWW 2005*, pages 442–452.

Michael J. Cafarella, Christopher Re, Dan Suciu, and Oren Etzioni. 2007. Structured Querying of Web Text Data: A Technical Challenge. In *Proceedings of CIDR 2007*, pages 225–234.

Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences*. Psychology Press.

William H. Fletcher. 2007. Implementing a BNC-Compare-able Web Corpus. In *Proceedings of the 3rd Web as Corpus Workshop*, pages 43–56.

Catherine O. Fritz, Peter E. Morris, and Jennifer J. Richler. 2012. Effect Size Estimates: Current Use, Calculations, and Interpretation. *Journal of Experimental Psychology: General*, 141(1):2.

Andrew Kehoe and Antoinette Renouf. 2002. WebCorp: Applying the Web to Linguistics and Linguistics to the Web. In *Proceedings of WWW 2002 (Posters)*.

Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2010. *Research Methods in Human-Computer Interaction*. Wiley Publishing.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers.

Jean-Baptiste Michel, Yuan K. Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez L. Aiden. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331(6014):176–182.

Raphael Mudge. 2010. The Design of a Proofreading Software Service. In *Proceedings of HLT 2010 Workshop on Computational Linguistics and Writing*, pages 24–32.

Davood Rafiei and Haobin Li. 2009. Data Extraction from the Web Using Wild Card Queries. In *Proceedings of CIKM 2009*, pages 1939–1942.

Nornadiah Mohd Razali and Yap Bee Wah. 2011. Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests. *Journal of Statistical Modeling and Analytics*, 2(1):21–33.

Philip Resnik and Aaron Elkiss. 2005. The Linguist's Search Engine: An Overview. In *Proceedings of ACL 2005 (Posters and Demos)*, pages 33–36.

Patrick Riehmann, Henning Gruendl, Bernd Froehlich, Martin Potthast, Martin Trenkmann, and Benno Stein. 2011. The NETSPEAK WORDGRAPH: Visualizing Keywords in Context. In *Proceedings of PacificVis 2011*, pages 123–130.

Patrick Riehmann, Henning Gruendl, Martin Potthast, Martin Trenkmann, Benno Stein, and Bernd Froehlich. 2012. WORDGRAPH: Keyword-in-Context Visualization for NETSPEAK's Wildcard Search. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1411–1423.

J. Sachs, P. Tung, and R.Y.H. Lam. 1997. How to Construct a Cloze Test: Lessons from Testing Measurement Theory Models. *Perspectives*, 9:145–160.

Satoshi Sekine. 2008. A Linguistic Knowledge Discovery Tool: Very Large N-gram Database Search with Arbitrary Wildcards. In *Proceedings of COLING 2008 (Demos)*, pages 181–184.

Benno Stein, Martin Potthast, and Martin Trenkmann. 2010. Retrieving Customary Web Language to Assist Writers. In *Proceedings of ECIR 2010*, pages 631–635.

W. L. Taylor. 1953. Cloze Procedure: A New Tool for Measuring Readability. *Journalism Quarterly*, 30:415–433.

Dominic Tsang and Sanjay Chawla. 2011. A Robust Index for Regular Expression Queries. In *Proceedings of CIKM 2011*, pages 2365–2368.

Appendix

A Questionnaire A from Experiments 1 and 2

1. I really prefer just anything __________ watching television.
   × against ✓ to × about × on
2. Has Tony's new book __________ yet?
   ✓ come out × published × developed × drawn up
3. If this plan __________ off, I promise you you'll get the credit for it.
   × lets × goes × gets ✓ comes
4. Helen had great admiration __________ her history teacher.
   × in × to ✓ for × on
5. I just couldn't __________ over how well the team played!
   ✓ get × turn × make × put
6. The problem stems __________ the government's lack of action.
   × out ✓ from × under × for
7. It's too late to phone Jill at work, at any __________ .
   × case × time × situation ✓ rate
8. I'm afraid I'm not very good __________ children.
   × about × for ✓ with × at
9. We are __________ no obligation to change goods which were not purchased here.
   × with ✓ under × to × at

B Questionnaire B from Experiments 1 and 2

1. Don't worry about the lunch. I'll __________ to it.
   × look × prepare × care ✓ see
2. I am afraid that these regulations have to be __________ with.
   × provided ✓ complied × faced × met
3. Our thoughts __________ on our four missing colleagues.
   × based ✓ centred × laid × depended
4. Carol doesn't have a very good relationship __________ her mother.
   ✓ with × at × for × to

5. It seems to be your boss who is __________ fault in this case.
   × under × with ✓ at × for
6. Being rich doesn't count __________ much on a desert island.
   × on × to × of ✓ for
7. The policeman __________ me off with a warning as it was Christmas.
   × sent × gave ✓ let × set
8. Tina is an authority __________ Byzantine architecture.
   ✓ on × for × with × in
9. I was __________ the impression that you liked Indian food.
   × at × with × of ✓ under

C Questionnaire A from Experiment 3

Choose the word which fits best using the options operator `[<words>]`.

1. If you spend so much money every day, you will __________ out of money before the end of the month.
   × pay × use ✓ run × take
2. You need to take __________ all your other clothes before you put on your swimming costume.
   × down × away × out ✓ off
3. I'm afraid I'm not very good __________ history.
   × about × for ✓ at × with

Choose the best synonym for the underlined word using the synonym operator ∼`<word>`.

4. I love studying geometry the most.
   × hate × absent ✓ enjoy × difficult
5. My ambition is to become a computer scientist.
   × thought × reward × study ✓ dream
6. Your action will have serious consequences.
   ✓ effects × events × reasons × affects

Choose the correct word order using the word order operator `{<words>}`.

7. The __________ bird! I'm going to help it!
   ✓ poor little × little poor
8. She was wearing a __________ dress.
   × green beautiful ✓ beautiful green
9. I plan on wearing my __________ coat.
   ✓ long black × black long

D Questionnaire B from Experiment 3

Choose the word which fits best using the options operator `[<words>]`.

1. Sometimes Julia speaks very quickly so the other students have to ask her to slow __________ .
   ✓ down × up × out × off
2. The missing plane has apparently disappeared without a __________ .
   × sign × news × word ✓ trace
3. When Gabriel's credit card stopped, he cut it __________ many small pieces.
   × out ✓ into × apart × in

Choose the best synonym for the underlined word using the synonym operator ∼`<word>`.

4. I choose to study the differences between alligators and crocodiles.
   × make × buy ✓ prefer × wash
5. I cannot find my money. Can you get me my billfold?
   ✓ wallet × pocket × watch × bag
6. This is a very rough environment for elephants to live in.
   ✓ harsh × abrasive × coarse × beneficial

Choose the correct word order using the word order operator `{<words>}`.

7. She sold the __________ chairs at a yard sale.
   × wooden old ✓ old wooden
8. The __________ years were fantastic.
   × two first ✓ first two
9. It's close to the __________ building.
   ✓ big blue × blue big