

Automatic Reformulation of Children's Search Queries

Maarten van Kalsbeek, Joost de Wit, Dolf Trieschnigg,
Paul van der Vet, Theo Huibers and Djoerd Hiemstra
University of Twente

Abstract

The number of children that have access to an Internet connection (at home or at school) is large and growing fast. Many of these children search the web by using a search engine. These search engines do not consider their skills and preferences however, which makes searching difficult. This paper tries to uncover methods and techniques that can be used to automatically improve search results on queries formulated by children. In order to achieve this, a prototype of a query expander is built that implements several of these techniques. The paper concludes with an evaluation of the prototype and a discussion of the promising results.

General Terms

Algorithms, Human Factors, Design

Keywords

Automatic Query Expansion, Search Behavior,

1 Introduction

According to a survey conducted in 2003 by the U.S. Census Bureau, 56.1% of the children between 3 and 17 years old are using the Internet in some way [1]. The number of children that has access to an Internet connection (at home or at school) is even larger and growing fast. Although 66.1% of the children uses the computer at home for school assignments and 64.4% just to surf around and search for all kinds of information, finding what they were looking for remains a big problem because search tools do not consider their skills and preferences.

This paper presents a research performed in order to uncover methods and techniques that can be used to automatically improve search results on queries formulated by children. The paper starts with a short overview of research performed on related topics. Then the research questions are presented, followed by the research method.

Section 4 discusses the characteristics that are specific to queries formulated by children, after which section 5 introduces potential solutions to these discrepancies. In section 6 the design and implementation of our system is explained. The evaluation method and results are mentioned in sections 7. The research is finally concluded in Section 8.

This report is published as CTIT Technical Report 10-X, *University of Twente*, 2010

2 Related Work

Although the number of children using and searching the Internet is growing rapidly, little effort is made to ease their struggle to find information. There are some search engines specifically designed for children, but most of them do not cope with the types of errors children make when formulating their queries. Some of these search engines will be described in the following section, followed by a description of research related to (automatic) query expansion and the search behavior of children.

2.1 Search Engines for Children

The fact that there is a need for search engines specifically targeted at children is recognized by companies such as Yahoo! and Google. They offer Yahoo! Kids [5] (formerly known as Yahoo!igans!) and Google SafeSearch [9] respectively. But there are other popular search engines for kids like Ask for Kids [10] (once Ask Jeeves), NetNanny [11], KidsClick! [12] and NetWijs [13] (in Dutch). These search engines are not so different from regular search engines however. The only main difference is that they try to retrieve only documents suited for children by filtering out adult contents or by limiting the corpus to suited documents. They do not take into account the way children perform their searches, nor do they anticipate mistakes children may make. Most of the search engines offer only the possibility to enter a Boolean query and they do nothing to correct errors made by the children. However, children do tend to make a rather large amount of (spelling) errors when formulating queries [5]. Research also shows that children do not tend to use (manual) Boolean logic, as it is hard to formulate [5].

2.2 Automatic Query Expansion

According to literature one of the main causes of retrieval failures is the mismatch between terminology used in a user's query and document contents [15]. Query expansion is a well known technique to overcome this discrepancy. With query expansion the user's query will be expanded with terms related to those used in the query. Finding these related terms however is really difficult because the average length of a search query is less than three words [14]. According to Gauch et al. there are three main sources for related words which vary in their level of specificity. The sources they have defined are: query specific; corpus specific; and language specific. Query-specific terms can be identified by locating new terms in a subset of the documents retrieved by a specific query. Corpus-specific terms are found by analyzing the contents of a particular full-text database to identify terms used in similar ways. Language-specific terms, on the other hand, may be

found from generally available online thesauri which are not tailored for any particular text collection. We mainly adopted the third class for our query expander.

2.3 Children’s Search Behavior

Search behavior of children has been an active field of research over the last decades. Several studies have investigated children’s search behavior on computer systems, ranging from on-line catalogs to the Internet [3]. Our design decisions for the query expander prototype were merely based on research performed in the area of information processing and motor skills and searching and browsing skills. Studies in these areas show that children have difficulties using a mouse or keyboard and find it hard to formulate and modify search queries due to a lack of domain knowledge [7]. Other important findings are that children get frustrated when their search fails and that they have difficulties with spelling and Boolean logic [3][4][5]. Unfortunately none of the research papers described how children’s queries were formulated and the type of spelling errors that were made.

3 Research Questions and Method

The aim of this research project is to design and implement a prototype of a query expander, allowing us to investigate a variety of methods and techniques to automatically improve search results on queries formulated by children. The overall research question therefore is:

“How can we build a query handler that can automatically improve search results by transforming search queries formulated by children?”

In order to answer this question we try to find answers to the following sub questions:

1. Which aspects characterize queries formulated by children? And how do these characteristics influence the search results?
2. Which methods and techniques are available that will most likely improve the search results, in relation to the aspects defined in sub question 1?
3. Which of these methods and techniques have the highest probability of improving the search results?
4. How can a method or technique be evaluated?
5. How does each of the chosen methods and techniques affect the search results either combined or by itself?

Given the time constraints of this research it is necessary to narrow its scope. Firstly, we only focus on handling queries formulated by children aged 4 to 16. Secondly, interface aspects are not within the scope of the project. And finally, this research focuses on the Dutch language, although most of it will also be applicable to English as well.

This research will in part be founded on literature, and in part on our own research findings. The first two sub questions will mainly be answered by looking at relevant literature. Our own insights into the subject will also be incorporated. This also holds true for the third sub question which we try to answer by reasoning which methods are most worthy of being implemented. These first three questions will form the basis for our query handler.

By answering sub questions four and five, we try and provide a sound evaluation of these techniques. We will argue what way makes sense to evaluate the implemented techniques and methods.

The answer to the last sub question will eventually provide us with the data needed to answer our main research question. Section 7 will provide more details about how this evaluation is carried out.

4 A Child’s Query Characteristics

In order to be able to improve search queries formulated by children we drew up a list of possible error/difference classes. Each of these classes represents a way in which children write different from common Dutch. We have consulted a primary school teacher to help us constructing this list and performed a small literature study [20][21][22]. The twenty classes we have identified are listed in Table 1.

<i>Class</i>	<i>Examples</i>
Typing errors	Elephant, expansion
Verbal spelling	Sizzors, stapeler
Plural phonetic transcription	BB (bees)
Number words	W8, 4u, xs4all
Slang	Chill out, dingbat, flakey
Descriptional writing	Karen, Kristel & Kathleen (K3)
Abbreviations	Info (information)
Short words	Kath (Kathleen)
Acronyms	AM (ante meridiem)
Combined Short words	Biotech <-> Biological technology
No vowels	hll wrld (hello world)
Special characters	€pe (Europe)
Smileys	:), ;-)
VIP-emoticons	:-()B (Pamela Anderson)
Hypernyms	Dog -> Animal
Hyponyms	Animal -> dog
Meronyms	Knob -> Door
Holonyms	Door -> Knob
Synonyms	Children <-> Kids

Table 1: Children’s difference classes

Most of these classes are self-explanatory in combination with the examples given. Some of these classes may still need some clarification. They will be explained in the next paragraphs.

Plural phonetic transcription does not appear to occur that often in English. In Dutch however, it does. Similar to e.g. number words, abbreviations and short words, it originates from cellular messaging. The class is made up combinations of a double consonant like “bb” which can be pronounced as (two) bees. The most common Dutch form is “ff” for “effen” (which is in turn a degenerated term for “even”).

Short words are alike abbreviations. However there’s a subtle difference. An abbreviation usually only has a single form, whereas a short word may be formed in different ways by

different persons. It is commonly also less known. The “combined short words” class comprises short words consisting of multiple “short words”.

In case of a hypernym, a child will be searching for a specific occurrence of something, like “dog” instead of “animal”. In case of a hyponym it’s the other way around. A meronym means (s)he’s searching for a part of something, like “tail” instead of “dog”. Finally, holonyms is the other way around where a child is searching for “dog” instead of e.g. “paw”.

The identified classes form the basis for a set of solution classes which will be described in the following section.

5 Potential Solutions

Now that we have defined the ways in which a child’s query can differ from document contents, we have to find means to close these gaps. We have based our possible solutions on literature and known concepts. The possible solutions we have identified can be classified into the following groups:

1. Dictionary lookup
2. Phonetic algorithm
3. Thesaurus or WordNet [23][24]
4. Clustering
5. Auto complete
6. Spelling correction

In a dictionary lookup you take some kind of dictionary $D: T \rightarrow S_T$ that maps term T onto a set S_T that contains zero or more terms and expand all terms in the query according to this mapping. When D contains the relation “AM” \rightarrow {“ante meridiem”} for example the query “12 AM” will be expanded to “12 AM ante meridiem”. Expanding queries this way is easy and fast, but it will not take the context of the query into account.

A phonetic algorithm is an algorithm for indexing of words by their pronunciation. It encodes words that are pronounced in the same way to the same phonetic code. There are several well-known and widely used algorithms like Soundex, Methaphone and the more accurate Double Methaphone. Unfortunately these algorithms were developed for use with the English language.

The third potential solution is the use of a thesaurus or WordNet which is an ontology of words with similar, related, or opposite meanings. The use of such a language-specific source for query expansion is described by Gauch in [14].

The extraction of related terms from the top ranked documents retrieved by a specific query, and expanding the original query with these terms is the fourth possible solution we have identified. We use the term clustering for this process in order to distinguish it from the expansion of the query with terms obtained through other means.

Auto complete is the process of expanding a query term T with terms S_T that satisfy:

$$(\forall x \in S_T : stratsWith(T, x))$$

So the query containing T gets expanded with all terms from the lexicon that start with that term as a prefix.

Spelling correction tries to resolve typographic errors such as insertions, deletions, substitutions, and transpositions of letters which are common mistakes made by children because their motor skills are not fully developed yet [3][4].

How the possible solutions we have identified map onto the difference classes specified earlier is shown in Table 2.

<i>Class</i>	<i>Possible solution</i>
Typing errors	2, 6
Verbal spelling	2
Plural phonetic transcription	2
Number words	1, 2
Slang	1
Descriptive writing	4
Abbreviations	1, 5
Short words	5
Acronyms	1, 4
Combined Short words	1
No vowels	1
Special characters	1, 2
Smileys	1
VIP-emoticons	1
Hypernyms	3
Hyponyms	3
Meronyms	3
Holonyms	3
Synonyms	3

Table 2: Mapping possible solutions onto difference classes

How we have implemented these possible solutions and how these implementations make up the query expander is described in the following section.

6 System Design and Implementation

In order to evaluate the effects of automatic expansion of a child’s query we have implemented the solutions explained in section 5 so that they cover most of the difference classes that were described in section 4. We have built our query expander on top of the information retrieval system Terrier [15]. The Terrier project started in 2000 at the University of Glasgow and aims to provide a state-of-the-art test-bed for research and experimentation. It provides developers with a set of API’s through which Terrier can be extended.

The design of the query expander together with Terrier is outlined in Figure 1. The query expander consists roughly of four components: the index manager, a preprocessing pipeline, parser and the query pipeline. The following subsections will describe these components in more detail.

6.1 Indexing

Terrier builds all the index structures (document index, lexicon, etc.) necessary for fast document retrieval and makes them accessible through an API. Our Index Manager uses the lexicon generated by Terrier to create some additional index structures like a vowelless index and a phonetic index. These indices can be used to speed up the expansion process that mainly takes place in the query pipeline.

6.2 Preprocessor

The preprocessor is implemented as a pipeline that consists of multiple pipeline objects. Each pipeline object processes the string that is sent through the pipeline in a different way.

The EmoticonTranslator object for example replaces the

emoticons in the query by their respective meaning, while the SpecialCharacterTranslator translates special characters like € and *. This preprocessing step is necessary in order to make the query fit in the grammar specified by Terrier.

6.3 Parser

The parser works quite straightforward, it parses the preprocessed query and returns an Abstract Syntax Tree (AST) representing that query. The terms in the AST are accessed through a DepthFirstAdapter that can be subclassed in order to perform special actions on different types of nodes.

6.4 Query pipeline

The AST that was created by the parser is then sent through the query pipeline. The query pipeline can be built dynamically by combining zero or more pipeline objects. Each pipeline object expands the incoming AST in a special way and passes the result on to the next pipeline object. We have implemented twelve different expanders as shown in Table 3.

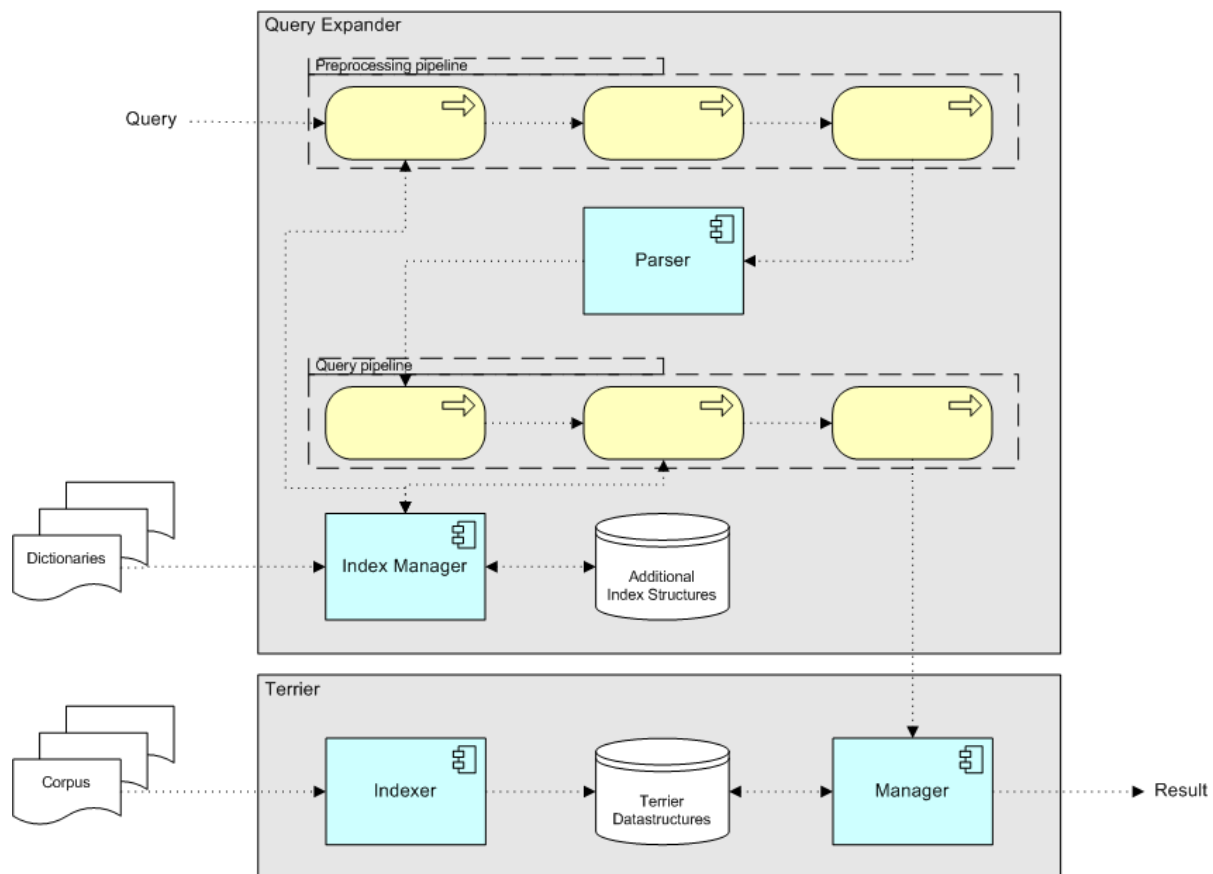


Figure 1: System Design

#	Expander	Solution class
1	NumberWordExpander	1
2	SpellingCorrection	6
3	ShortWordExpander	5
4	PhoneticExpander	2
5	VowelLessWordExpander	1
6	PluralPhoneticExpander	1
7	AcronymExpander	1
8	WordnetSynonymExpander	3
9	WordnetHolonymExpander	3
10	WordnetHyperonymExpander	3
11	WordnetHyponymExpander	3
12	WordnetMeronymExpander	3

Table 3: Implemented expanders

NumberWordExpander

The NumberWordExpander takes a query and expands it with the translations of all numberwords (words that contain one or more numbers) in the query. “I’ll w8 4 you” for example is expanded to “I’ll w8 weight wait 4 four for you” because the dictionary used by the NumberWordExpander contains the entries “4” → {“four”, “for”} and “8” → {“eight”, “ait”}.

SpellingCorrection

The SpellingCorrection object resolves typographic errors by calculating the Levenshtein distance [19] against the words in the lexicon and expanding the query with those words that have the smallest distance. The Levenshtein distance is the smallest number of insertions, deletions, and substitutions required to change one string into another. The query is only expanded when it contains terms that are not present in the lexicon; otherwise it assumes that all the words are spelled correctly. The SpellingCorrection object has two important parameters: the maximum allowed Levenshtein distance and the minimum length of a term. A misspelled term is only expanded when its length is larger than or equal to minTermLength. When that occurs, the term is expanded to terms in the lexicon for which the Levenshtein distance is smaller than or equal to maxAllowedDistance.

ShortWordExpander

The ShortWordExpander is an implementation of the auto complete solution described in section 5. It expands each term T in a query with terms S_T from the lexicon that satisfy:

$$(\forall x \in S_T : stratsWith(T, x))$$

So the query “fire” could be expanded to “fire firefighter” depending on the lexicon.

PhoneticExpander

The PhoneticExpander expands the query with terms that are pronounced in the same way as those included in the query. At application startup a phonetic index is created that maps all words in the lexicon onto their corresponding phonetic code using a phonetic algorithm such as Soundex, Methaphone or Double Methaphone. The PhoneticExpander encodes the query terms and then uses the phonetic index to efficiently find words in the lexicon that are pronounced in a similarly and expands the query with those words. We have used the Double Methaphone in our implementation because it should produce more accurate results than Soundex and Methaphone.

VowelLessWordExpander

The VowelLessWordExpander expands queries that contain vowelless words with words from the lexicon that consist of the same consonants. So “xpnsn” could be expanded to “xpnsn expansion”. The expander uses an index structure that is created at application startup. This index consists of all the words from the lexicon with their vowels removed and makes the execution of the expander more time efficient.

PluralPhoneticExpander

The PluralPhoneticExpander does to characters what the NumberWordExpander does to numbers. It would expand the query “bb” into “bb bees” for example.

AcronymExpander

The AcronymExpander loads a dictionary that maps acronyms onto their meaning. This dictionary is then used to lookup the acronyms in the query so they can be expanded. The query “NATO” will be expanded to “NATO North Atlantic Treaty Organisation” for example.

WordNet Expanders

In order to expand a query with synonyms, holonyms, hyperonyms, hyponyms and/or meronyms of the terms included in that query, we have used a Dutch version of the semantic lexicon WordNet. We have built a separate expander for each of the semantic relations. Those expanders lookup all the terms in the query in WordNet and expand the query with the related words of the selected type.

7 Evaluation

To determine which query pipelines, consisting of zero or more expander objects, improve the search results on queries formulated by children, we used the prototype to expand a list of queries. These queries were based on 17 different scenarios varying from a child looking for information about dogs to a kid who is interested in the size of the sun. For each of these scenarios we have created six queries that are either spelled correct or include the mistakes we have identified in section 4. This resulted in 102 queries that needed to be expanded.

We have expanded those queries in 18 different ways by applying each of the query expanders separately and by combining some of the expanders in a way that we thought to be promising. The 18 evaluation classes we defined are listed in Table 4.

#	Query pipeline
0	none (preprocessing only)
1	NumberWordExpander
2	SpellingCorrection
3	ShortWordExpander
4	PhoneticExpander
5	VowelLessWordExpander
6	PluralPhoneticExpander
7	AcronymExpander
8	WordnetSynonymExpander
9	WordnetHolonymExpander
10	WordnetHyperonymExpander
11	WordnetHyponymExpander
12	WordnetMeronymExpander
13	SpellingCorrection PhoneticExpander
14	SpellingCorrection NumberWordExpander VowelLessWordExpander PluralPhoneticExpander AcronymExpander
15	SpellingCorrection NumberWordExpander VowelLessWordExpander PluralPhoneticExpander AcronymExpander ShortWordExpander
16	NumberWordExpander ShortWordExpander PluralPhoneticExpander PhoneticExpander
17	WordnetHyperonymExpander WordnetMeronymExpander WordnetSynonymExpander

Table 4: Evaluation classes

Expanding the 102 queries in 18 different ways resulted in 1836 new queries. To decide whether these queries really improved we have classified them according to the words that were added to the query. The result classes were:

1. Only relevant words have been added
2. Mainly relevant words have been added
3. The number of relevant and irrelevant words that have been added is about the same
4. Mainly irrelevant words have been added

5. Only irrelevant words have been added
6. The query is not expanded at all

In order to categorize the expanded queries in a fair and unbiased way we constructed a panel of 32 people. The panel categorized the queries through a specially designed website that presented each member of the panel seven randomly chosen scenarios with eight randomly chosen queries. The website administered who had categorized which queries and prevented a member from categorizing the same query twice. Panel members could categorize multiple sets of seven times eight queries.

After each query was categorized at least twice we closed the website and started collecting the data.

7.1 Results

The evaluation panel has provided us with a total of 2672 query evaluation results. Because in total only 672 queries were found to be expanded, this results in an average coverage of 3.98. Thus, on average each expanded query has been classified by four persons, giving our results a good reliability.

The evaluation results per evaluation class as listed in Table 4 are shown in Figure 2. Note that this figure only shows the results for the expanded queries, as the amount of non-expanded queries does not tell you anything about how a query is expanded.

Detailed results per evaluation class can be found in Appendix A. These figures will also show you in how many cases the query was evaluated as being expanded or not.

The results provide us with some interesting details which allow us to fulfill the goal of our research. In the next subsection we argue which techniques appear most promising in regard to query expansion for children. This will be followed by some notes about the techniques that did not yield promising results. Finally, this section ends with some general notes in regard to our research.

7.2 Promising techniques

When looking at Figure 2 it is obvious to see which technique is most promising to implement. The WordNet Synonym Expander only adds correct words to a query in over 93% of the cases according to our evaluation. In the remaining 7% it still adds more relevant words than irrelevant words. This means that whenever the expander expands a query (which occurs in 14% of the cases, see Appendix B) it is highly reasonable to assume search results will be improved. Because literature also shows children will usually not try searching for synonyms when their query fails it makes sense to implement this technique into any search engine tailored for children.

Another technique scoring quite high is the Phonetic Expander. It does score less than the Synonym Expander. A reason for this can be found in the fact that it has a lot more variables influencing the results. When taking a deeper look at the results it was found to score bad mainly on words consisting of only a few characters or phonemes. An example is the word “Thea” (from Thea Beckman, a Dutch author) which was expanded to “1th th th1 th6 thea theia theo thi thu”. Unfortunately, because of time constraints we were not able to put any effort into fine-tuning each expander after evaluation. We do believe that any fine-tuning in a real-world application would yield very high results.

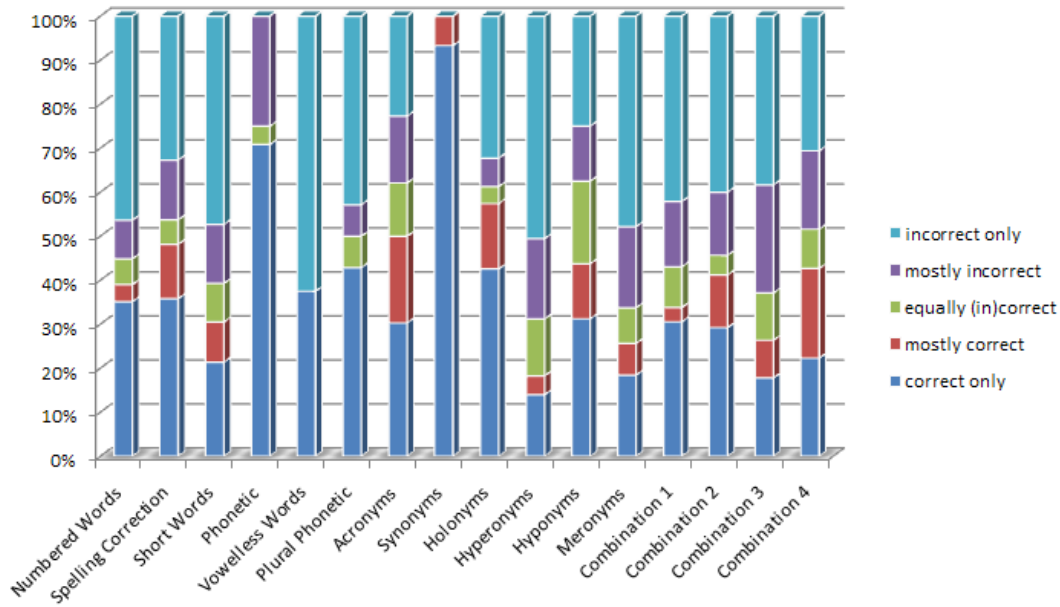


Figure 2: Evaluation results per query pipeline

7.3 Not so promising techniques

There are also a number of query pipelines that are doing more harm than good to the original search query.

The first one to note is the Hyperonym Expander. We find the bad performance of this expander to be an expectable outcome. When a child is looking for a hyperonym of a word (s)he used in his/her query it is likely the correct will be added. However, in most cases it is possible to find a hyperonym. As the expander can not decide if a child is actually looking for a hyperonym, it will almost always expand a query, even when it shouldn't. Because of this, in most cases the expander will only worsen the search results. The expander also does not allow for much fine-tuning, as it will simply provide a complete list of the available hyperonyms.

Because we were not able to perform any fine-tuning to the expanders after the evaluation was performed, most expanders currently roughly expand the query correctly as many times as they expand it incorrectly. We believe that in most cases, except for the WordNet (-onym) expanders, fine-tuning will be able to improve the results significantly. Fine-tuning will also greatly improve the results of the combined expanders as in general they will suffer a lot from only a single bad expander.

7.4 Other notes

First of all, it should be noted that our implementations are highly language dependent. Because we have only been focusing on the Dutch language it's hard to say if and in what way results will differ for other languages like English. In some cases we think results will be a lot better for especially the English language. This especially holds true for our implementation of the phonetic expander as phonetic translators are language dependent and for English they are simply a lot more mature and well-spread.

Second it's also important to note that in other cases the results will also depend on our implementation of an expander. In some cases other methods that we did not examine may be available,

providing better results. This is also related to the fact that because of time-constraints we were not able to perform any fine-tunings after our evaluation. We do, however, not feel this should pose a problem. Our research is meant to only give insights into which type of expanders appear promising. The results are simply meant to provide a direction for future research and should only be seen as early research into the possibilities.

Third, we wish to point out that whenever a search engine starts to automatically alter a search query, it's important to protect children from less preferable results. An example of this was found in our evaluation. We defined a scenario where a child was looking for information about the weather, and snow in specific. One of our combination expanders decided to translate the query for the Dutch word "sneeuw" ("snow" in English) into "dope hard drug hard-drug harddrug sneeuw". A solution for this would be the usage of a stop-word list of words a query can not automatically be expanded to. Another possibility is to constrain the document collection to documents that are suitable for the intended age group. In the last case the query would still be expanded but a result about e.g. "hard drug" would not be found.

8 Conclusion and Future Work

Our results have yielded some interesting insights into the problems children may be experiencing when creating a search query. In addition, it also provides insights in the methods and techniques that are available to overcome these problems. We have found some techniques to almost only alter search queries in a promising way. On the other hand, it was also found that some problems can not be overcome with the methods and techniques evaluated in our research.

We believe our research provides a good foundation for future work. It can give a sense of direction as to which techniques deserve further research attention, and which ones do not. Therefore we feel confident to say we were able to answer our main research question and meet the goal we have set.

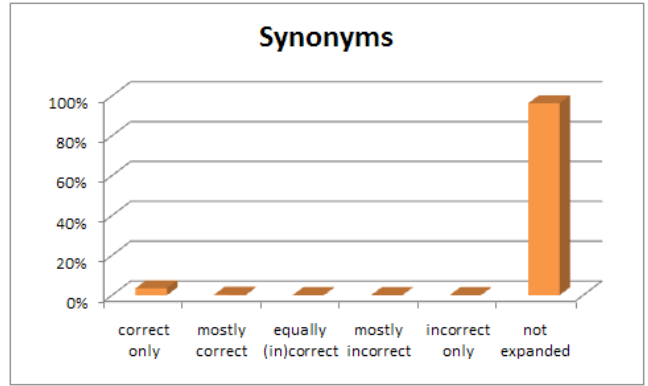
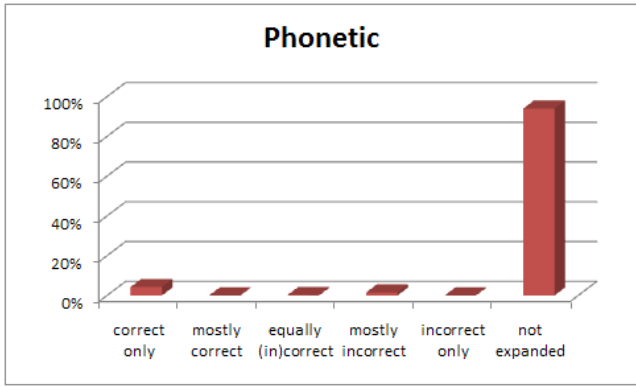
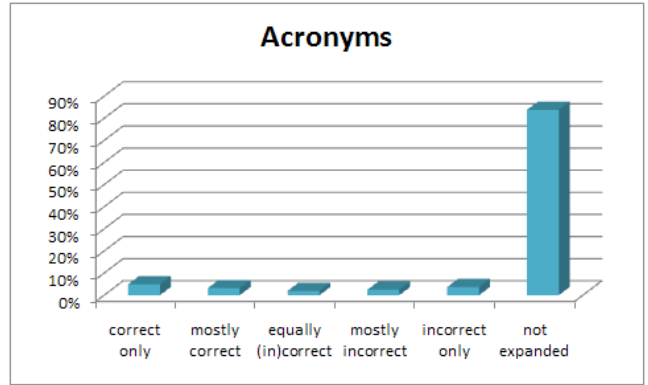
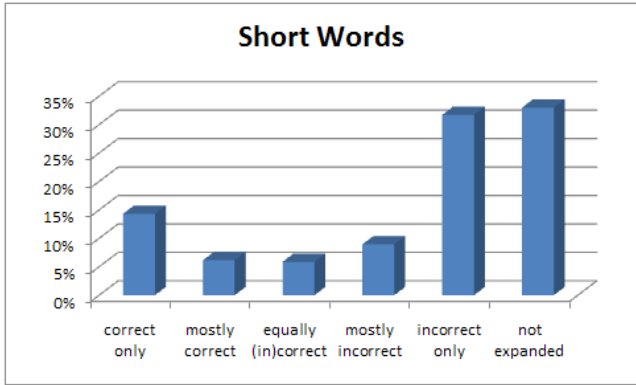
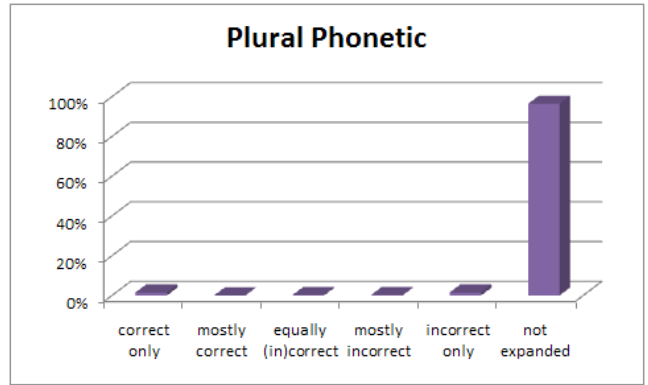
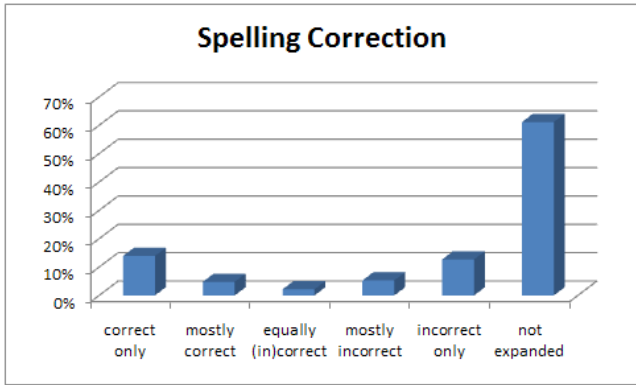
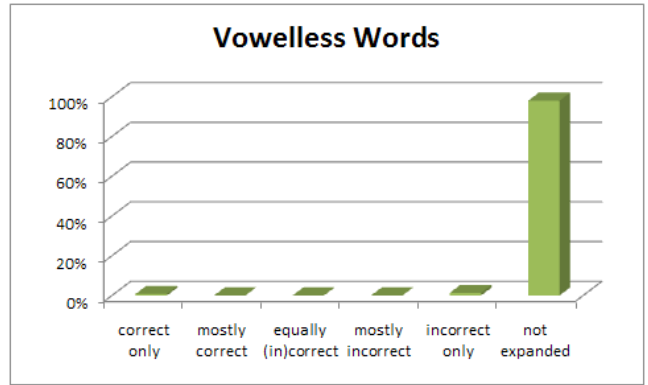
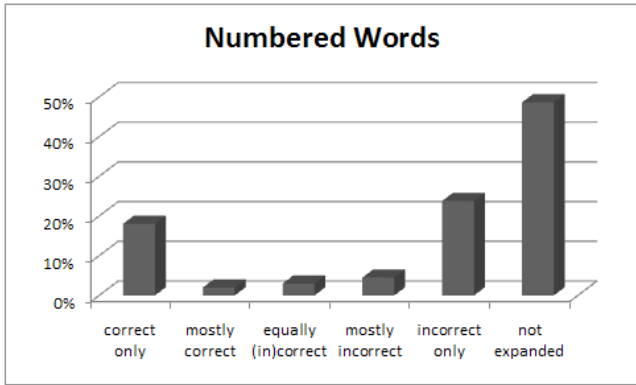
Acknowledgments

This research is funded in part by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 231507

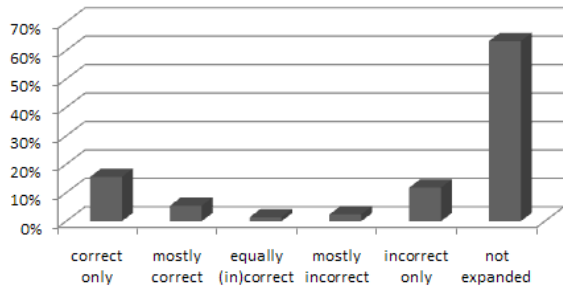
References

- [1] Day, J., Janus, A., et al. (2005). Computer and Internet Use in the United States: 2003. U.S. Census Bureau, October 2005.
- [2] Borgman, C., Hirsh, S., et al. (1995). Children's Searching Behavior on Browsing and Keyword Online Catalogs: The Science Library Catalog Project. *JASIST*, 46 (9), 663-684.
- [3] Hirsh, S. (1999). Children's Relevance Criteria and Information Seeking on Electronic Resources. *Journal of the American Society for Information Science*, 50 (14), 1265-1283.
- [4] Hutchinson, H.B., Bederson, B.B., et al. (2006). The Evolution of the International Children's Digital Library Searching and Browsing Interface. In *Proceedings of the 2006 Conference on Interaction Design and Children*, 2006.
- [5] Hutchinson, H.B., Bederson, B.B., et al. (2006). Interface Design for Children's Searching and Browsing. In *Proceeding of the 2006 conference on Interaction design and children*, 2006.
- [6] Kuiper, E., Volman, M., Terwel, J. (2005). Internet als Informatiebron in het Onderwijs: een Verkenning van de Literatuur. In *Pedagogische Studiën*, 81 (6), 423-443.
- [7] Fidel R., Davies, R.K. (1999). A Visit to the Information Mall: Web Searching Behavior of High School Students. *Journal of the American Society for Information Science*, 50 (1), 24-37.
- [8] Yahoo! Kids (2010). *Kids Games, Kids Movies, Kids Music, and More - Yahoo! Kids*. Visited June 6, 2010. URL: <http://kids.yahoo.com>.
- [9] Google (2008). *Google SafeSearch*. Visited July 24, 2008. URL: <http://www.google.com/help/customize.html#safe>.
- [10] Ask for Kids (2010). *Ask for Kids*. Visited June 6, 2010. URL: <http://www.askforkids.com>.
- [11] NetNanny (2010). *NetNanny - Internet Filter, Parental Controls & Filter Software*. Visited June 6, 2010. URL: <http://www.netnanny.com>.
- [12] KidsClick! (2007). *KidsClick! Web Search*. Visited June 6, 2010. URL: <http://www.kidsclick.org>.
- [13] NetWijs (2007). *Netwijs -- Dé zoekmachine voor de basisschool*. Visited June 6, 2010. URL: <http://www.netwijs.nl>.
- [14] Gauch, S., Wang, J. (1999). A Corpus Analysis Approach for Automatic Query Expansion and Its Extension to Multiple Databases. *ACM Transactions on Information Systems*, 17 (3), 250-269.
- [15] Billerbeck, B., Zobel, J. (2004). Questioning Query Expansion: An Examination of Behaviour and Parameters. In *Proceedings of the Fifteenth Australasian Database Conference (ADC 2004)*, January 2004.
- [16] University of Glasgow (2010). *Terrier Information Retrieval Platform*. Visited June 6, 2010. URL: <http://ir.dcs.gla.ac.uk/terrier>.
- [17] Ounis, I., Amati, G. (2005). Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, 2005.
- [18] Ounis, I., Amati, G. (2005). Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2005.
- [19] Levenshtein, V.I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10 (8), 707-710.
- [20] Kennisnet, Centrum voor Mondiaal Onderwijs (2010). *WRLD HLL DG: N LSBRF VR SMSS*. Visited June 6, 2010. http://www.kennisnet.nl/vo/scholier_vmbo/perdagwijzer/sms/html/
- [21] Hilgers, L. (2006). *SMS-taal zkwrdnbk*. Visited December 11, 2006. URL: <http://www.sms-taal.nl>
- [22] Gendt, J., Weeda, R. (2005). Giggie: Internet Search Engine for Kids.
- [23] Vossen, P., Bloksma, L., Boersma, P. (1999). *The Dutch WordNet - Version 2, Final*. Visited February 4, 2007. URL: <http://www.vossen.info/docs/1999/DutchWordNet.pdf>
- [24] Princeton (2007). *WordNet - Princeton University Cognitive Science Laboratory*. Visited February 4, 2007. URL: <http://wordnet.princeton.edu/>

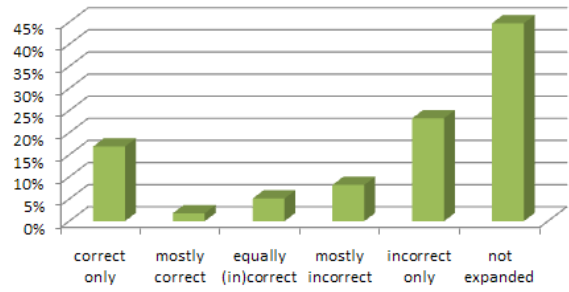
Appendix A: Detailed evaluation data



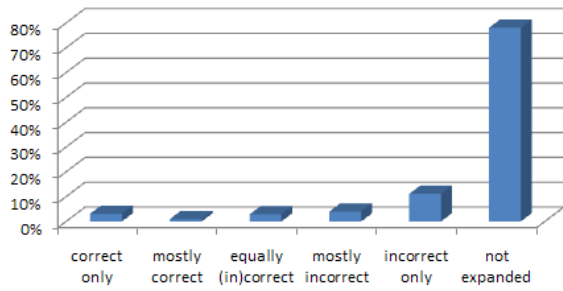
Holonyms



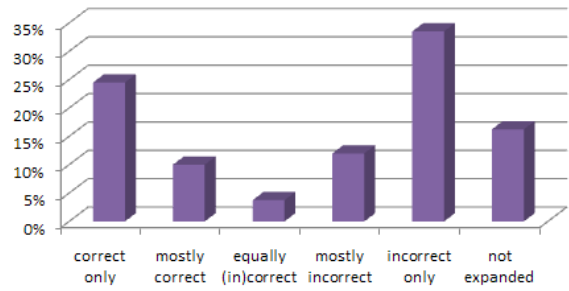
Combination 1



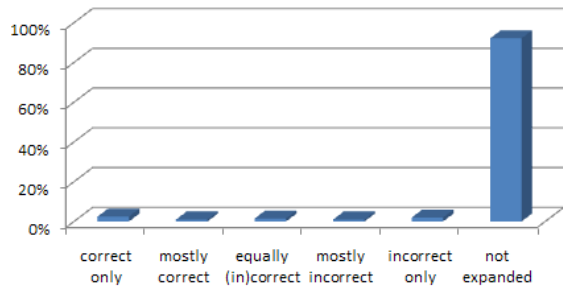
Hyperonyms



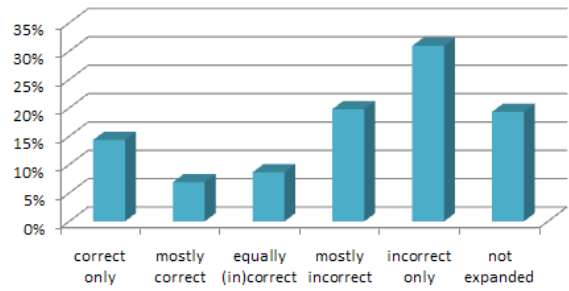
Combination 2



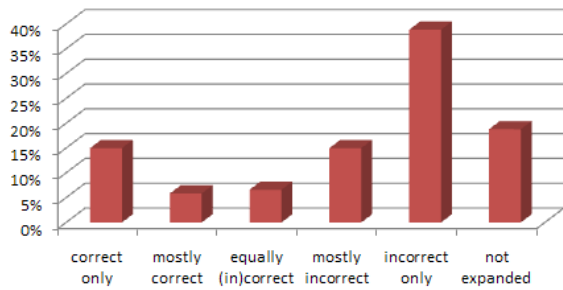
Hyponyms



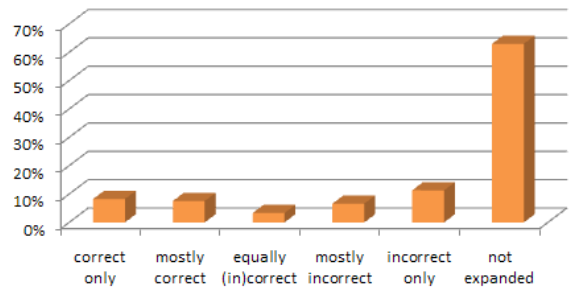
Combination 3



Meronyms



Combination 4



Appendix B:

Overview of the evaluation scenarios and queries

Spreekbeurt over de brandweer

Tim zit in groep 5 van de basisschool en moet een spreekbeurt geven over een zelf gekozen onderwerp. Tim heeft gekozen voor de brandweer omdat hij later graag brandweerman wil worden.

Query
Brandweer
Brandweerman
Brantweer
brandwer
brndwr
brandw

Spreekbeurt over het weer

Linda zit in groep 6 van de basisschool en ze gaat een spreekbeurt houden over het weer want dat leek haar vader wel een leuk onderwerp.

Query
weer
reegen
wint
wolken
sneeuw
sneuw

Spreekbeurt over Anne Frank

Tijdens een themaweek over de Tweede Wereldoorlog moet Léon een spreekbeurt over Anne Frank geven. Léon zit in groep 8 van de basisschool.

Query
Anne Frank
Ane Frank
Anne Frank huis
agterhuis
achterhuis
dagboek

Spreekbeurt over vliegtuigen

Maarten is 8 jaar oud en wil later graag piloot worden, net als zijn vader. Daarom houdt hij zijn spreekbeurt over vliegtuigen.

Query
Vliegtuig
Vkiegtug
Jumbojet
Pilot
KLM
Fliegen

Spreekbeurt over autoracen

Kevin (9) en Joost (10) zijn gek op autoracen en geven daarom samen een spreekbeurt over dit onderwerp.

Query
raceauto
race auto
reesauto
Formule 1
autoracen
f1

Spreekbeurt over de jaguar

Het lievelingsdier van Anieke is de jaguar. Ze heeft al een boekje over deze dieren en wil haar spreekbeurt graag over ze houden.

Query
jaguar
jaquar
jaguar dier
katachtigen
katachtige
kat8tige

Spreekbeurt over honden

Bij Bas thuis hebben ze 2 honden, een border terrier en een Friesche Stabij. Op school moet hij een spreekbeurt geven en hij kiest "honden" als onderwerp.

Query
hond
hodnen
hnd
Friesche Stabij
friese stabij
border terrier

Favoriete film

De nieuwe film van The Simpsons is net uit en Bart (8) wil er graag wat meer over weten.

Query
The Simpsons
the simsons
bart simpson
(_8-())
simpsns
smpsns

Zoekopdracht naar het milieu

De docent van groep 6 vindt het milieu erg belangrijk en de kinderen in zijn klas moeten er daarom een werkstuk over maken. Meindert is daarom op zoek naar geschikte informatie.

Query
het milieu
het mileu
millieu
broeikas
broekaseffekt
brksffkt

Zoekopdracht naar konijnen

Annemiek heeft van school uit de opdracht meegekregen om op te zoeken hoe oud konijnen kunnen worden.

Query
konijn
konjin
konjn
oud konijnen
konijgen
leftijd konijn

Zoekopdracht naar de ramadan

Tijdens de ramadan heeft het schoolTV weekjournaal een item over deze maand waarin moslims vasten. Geert (10) moet na de uitzending meer informatie over de ramadan zoeken.

Query
ramadan
rmdn
rammadan
ramadaan

rammaddan
ramad

Zoekopdracht naar de Europese Unie

Tijdens een themaweek over de Europese Unie krijgt Heleen de opdracht om uit te zoeken welke landen er allemaal lid van de EU zijn.

Query
europesche unie
europese unie
euroepse unei
ueropese nie
€pese unie
€pa

Boekbespreking over Kruistocht in spijkerbroek

Auke is gek op boeken van Thea Beckman en voor zijn boekbespreking op school zoekt hij meer informatie over haar beste boek, Kruistocht in spijkerbroek.

Query
Thea Beckman
Tea Beckman
Tea Bekman
Kruistocht in spijkerbroek
Krusitocht in spijkbroek
krstcht spkrbrk

Wat is er op TV?

Chris mag vanavond uitzoeken welke film ze thuis gaan kijken op TV. Deze kans krijgt hij niet vaak, dus wil hij graag een goede keus maken.

Query
vanavond films
tv gids
televisie
teeveegids
vanavond teevee
films van8

Informatie over de prinsjesdag

David heeft op school te horen gekregen dat ze het aanstaande dinsdag les over prinsjesdag krijgen omdat het dan de derde dinsdag van september is. Iedereen heeft de opdracht gekregen om in minimaal tien regels op te schrijven wat prinsjesdag nu

eigenlijk is. David heeft geen idee en wil daarom goede informatie vinden om aan z'n tien regels te komen.

Query
prinsjesdag
derde september
prinsjedag
prinsjesdach
prnsjsdgc
goudne koets

Informatie over ridders

Hans is helemaal gek van ridders. Z'n hele kamer hangt al vol met plaatjes en posters van ridders. Hans is op zoek naar nieuwe plaatjes van ridders om zijn collectie aan te vullen.

Query
ridders
ridderz
poostrs van ridders
riddes
middeleeuwen
middeleewen

De grootte van de zon

Floris' vader heeft hem verteld dat de zon wel 100 keer zo groot is als de aarde. Floris gelooft er echter niets van en besluit daarvoor bewijs te vinden.

Query
zon 100 keer zo groot
zon
aarde zon
wereld zon
zon 100x zo groot
son groter