# YouUndestood.me

YUM is an online environment built around a search engine, intended to make the searching process valuable for children. Yum does not pretend to build a new search engine, since studies \cite{ref} show that children tend to prefer popular search engines such as Google. Alternately, YUM acts as an intermediate layer between the child and an existing search engine in order to facilitate the communication between the two of them. For doing so, YUM puts into practice strategies oriented to amend issues popular search engines face when they are used by children, but also strategies targeted to enhance the search experience it in order to foster learning. A description of the mentioned strategies is provided below.

## Search intent

Children are unusually successful in formulating proper queries that search engines can understand\cite{Bil11}. Studies show that they tend to write long natural language queries, instead of the short keyword based queries a search engine usually expects \cite{Dru09}. Unfortunately, the longer the query the less probable is a search engine to retrieve proper results in response to it, a fact which can lead children to frustration, because of the inability to complete their information seeking task\cite{Dru09}. In addition, children also tend to misspell words, misspellings that can differ from the ones an average adult does. For example, children commonly repeat letters in a word to emphasize it, such as in “amaaaaaaaaazing”. In order to best satisfy children needs Yum takes advantage of QuIK\cite{Sven} a search intent module specifically designed for children. QuIK is able to address common patterns in queries written by children including but not limited to: diminutives, emphasis, children trendy terms or children specific misspellings. In doing so, QuIK transforms the initial children written query into a keyword based new query that captures the information expressed by the child in a way that can more easily be processed by search engines.

## Query suggestion

Even if the search intent module can fix and extract the main intent of a number of queries, there are decisions that QuIK cannot take on its own. Queries can be too short and therefore ambiguous, having different intent depending upon the audience, or can be too long and contain mixing information that can confuse the search engine in its retrieval process. In both cases, the user, as the main stakeholder, is the only one that knows the purpose of his search and can therefore improve the query in order to inform the search process involved in solving an information seeking task. For this reason, Yum includes ReQuIK \cite{requik} a query recommender that is also specifically tailored to children. This tool is able to provide suggestions that try to summarize the query using keywords in cases where it contained non valuable information, or even extend it in cases were the query was too short and ambiguous, potentially containing different meanings depending on the point of view of the user. Those suggestions are given to the user, so that he can pick the one that best fits his information requirements. ReQuIK is a multi-criteria query recommendation system that informs the recommendation process with traits commonly associated with children to suggest queries that (1) are associated with children topics, (2) will potentially retrieve resources with low levels of readability, and (3) are diverse enough to capture the different topics children can be interested at.

## Filtering by readability

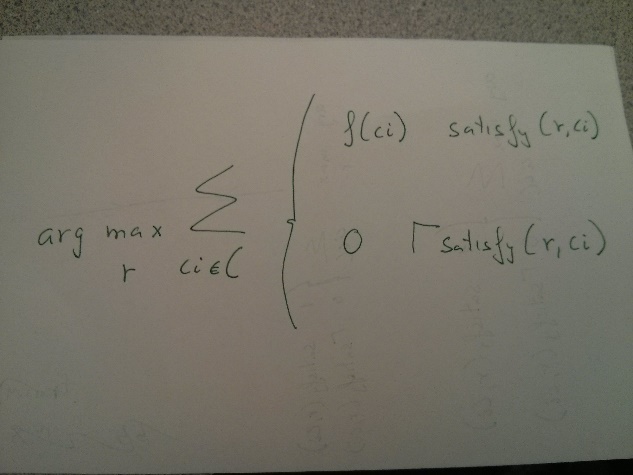
Even when the search engine has understood the intent of a children query and retrieves results that are related to the information the child is looking for, adequacy of retrieved resources is still not assured. K-12 students, tend to have difficulties understanding documents containing complex or technical vocabulary. For example, in the case where a child is looking for information about chemistry, retrieving a scientific publication would not be adequate, while retrieving information from an elementary chemistry book would. On cases where all retrieved documents are too complex, children may not succeed in their information seeking process, causing frustration to them. In order to avoid this situation, Yum takes advantage of a resource filtering strategy based on readability levels which ensures that the retrieved documents will be of the readability level required by each children. Users can pick a readability level at the same time they submit a query. Yum matches this readability level with the one predicted for retrieved results using the FleshKincaid \ref{fleshkincaid} (see formula XX) readability formula. Even if we plan to improve this formula in the future (see Future Work Section), we selected this formula as an initial point given that it is widely used for measuring readability of texts among educators nationwide and considered as a standard by institutions\ref{porposal31\_38\_40} to measure readability levels.

Placeholder for fleshkincaid

## Tracking

K-12 students have diverse reading abilities. Even in same grade class, students’ readability skills can differ. Furthermore, reading skills of each individual progressively update during the learning process. These facts make the strategy of one-fits-all not work in this contest, thus, there is a need of personalization of results. In Yum students have a personal account that enables them to give optional feedback on the resources they have retrieved and read while using the system, i.e. whether the resource was too “*easy”*, “*OK”* or “too *complex”*. This feedback is used to determine the reading skills for each user so that the system can retrieve documents adequate to that level of readability.

The problem of student’s readability prediction can be seen as a constraint satisfaction maximization problem, where, each feedback **f** produces a new constraint that needs to be satisfied by the readability of the student **s**, e.g. a student giving a feedback of “too hard” to a document of readability level 9 would generate the constraint readability(s) < 9. The predicted readability is the one that maximizes the amount of constraints satisfied. In case of multiple maximum values, the one that has the biggest distance to its two (lower and upper) closest constraints is selected.



Where **r** is the readability value, **ci** is an individual constraint among all **C** constraints for user **s**, and **f(ci)** refers to how new the constraint is, starting at 12 is the constraint was created at this month, and getting reduced by one if for every month, ending in 0 if the constraint was created more than 12 months ago.

Note that we also consider the time the resource was read by the student. This permits Yum to prioritize recent feedback, and allow the readability prediction adapt to the student over time.

Furthermore, in order to avoid the cold start problem, where the student does not have any constraint to satisfy. Every student account is initialized with two constraints based on the grade level of the student, those constraints are representative of one grade of deviation from the current grade of the student: readability(s) < grade(s)+1 and readability(s) > grade(s) -1. These constraints give a starting point to Yum, that will eventually be better adjusted when the student start using the environment.

# Yum for teachers

Children are not the only stakeholder of Yum, teachers can also obtain benefit of using it in the class environment. Working setting standards have changed from a vertical structure, where only the top individuals of the pyramid had to think critically and the lower parts just followed directions, to horizontal structure, where each individual is expected to collaborate with others and solve important problems using identification, searching, synthetizing, and communication skills \ref{leu13}. Given this change, nations worldwide have started developing education plans oriented to meet the new requirements of the current industry, such as the Common Core State Standards (CCSS) Initiative. CCSS claims educators to make an emphasis on higher level thinking during reading and writing and focus on the acquisition of skills such as research and comprehension using digital tools, i.e. as search engines\ref{leu13}. Furthermore, educational studies \ref{kni15} showcase the benefit of in class exercises such as exploratory tals, where students are asked to resolve a problem in groups discussing information found in resources obtained using a search engine. Unfortunately, teachers will never be able to propose such a task to their students and lead critical discussion, if students have problems using the search engine. Well because they cannot find the right query or because they are not able to understand the retrieved documents due to their complexity. YUm helps the teacher overcome those issues in class and focus on the critical discussion, rather than fixing individual problems of students that cannot find any result. Furthermore, YUm provides the teacher with monitoring tools that will serve to check how students are doing by reporting about the resources they have read and the feedback they gave about the complexity of them, so that he can give them assistance when needed. Therefore, we think that YUm will not only facilitate learning when children use it for their information discovery assignments at home, but it will also help the teacher in terms of reinforcing the learning in class.

@article{leu13,

title={The new literacies of online research and comprehension: Assessing and preparing students for the 21st century with Common Core State Standards},

author={Leu, Donald J and Forzani, Elena and Burlingame, Cheryl and Kulikowich, Jonna and Sedransk, Nell and Coiro, Julie and Kennedy, Clint},

journal={Quality reading instruction in the age of common core standards},

pages={219--236},

year={2013}

}

@article{knight2015role,

title={The role of exploratory talk in classroom search engine tasks},

author={Knight, Simon and Mercer, Neil},

journal={Technology, Pedagogy and Education},

volume={24},

number={3},

pages={303--319},

year={2015},

publisher={Taylor \& Francis}

}