D3.6 Terminology extraction for user modelling

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	3
Deliverable number	D3.6
Authors	Suzan Verberne, Maya Sappelli, Rianne Kaptein, Corné Versloot, John Schavemaker
Reviewers	Wessel Kraaij and Martijn Vastenburg
Date	2013-04-05
Version	3
Access Rights	SWELL Project Members
Status	Final
Deliverable type	Product

Document change log

Date	Name	Changes
2013-03-05	Suzan	Adapted document from 2012-Q4 version
2013-03-11	Suzan / Maya / John	Ready for internal review
2013-04-05	Suzan	Final version for deliverable

Related documents

D3.1a Functional design of the reasoning components in the SWELL@work tool

SWELL Partners:

Ericsson, NCSI, Noldus, Novay, Philips, TNO, Radboud Universiteit Nijmegen, Roessingh Research and Development, Universiteit Twente,

Summary

The goal of this deliverable is to identify the main descriptive terms for the working domain of a knowledge worker. A first approximation is to distil terminology from documents, authored by the person in focus. We evaluated and implemented a number of term extraction algorithm for which we found an optimal combination that we implemented in the SWELL tag cloud generator.

Contents

1 Introduction

Terminology plays an important role in SWELL. WP3 aims to develop tools that can assist knowledge workers in managing their information flow. When the information flow is not properly managed, this can result in feeling or being unproductive or inefficient. In order to help the user to select relevant and important information in the large body of incoming e-mails and online search results, we need to create a model of the user. A collection of domain-specific terminology is needed to create such a user model.

Most work on content-based user profiling is directed at personalization purposes, often in the context of search systems: if the search systems knows the interests of the user, it can adapt the search results so that they better suit the user. We aim to develop a user profile that not only can be used and interpreted by a system but also by the human user. The SWELL@work tool (see D3.1 functional design) exploits the user profile in order to estimate the importance of incoming information (e-mail and search results). In addition to that, we want to give the user insight in his/her profile: which terminology is important in which context, and which terminology is shared with co-workers?

The goal of this deliverable is to identify the main descriptive terms of the working domain offor a knowledge worker. The goal of terminology extraction (sometimes also called *terminology mining*, *term recognition*, or *glossary extraction*) is to automatically extract relevant terms from a given document collection. The main descriptive terms are saved in a term base.

The deliverable consists of three components, which have been implemented in Java:

- A back-end (‘TermThings’) in which several term extraction methods have been implemented
- An interface (‘SwellTagcloudBackend’) that transforms the output of the term extractor to a tag cloud format
- A front-end (‘UploadHTML’): a browser-based GUI for a demonstrator in which the user can upload documents and view the tag clouds that are generated by the backend

The demonstrator can be found and used at http://www.swell-project.net/publications/deliverables

2 Term extraction methodology

In tag clouds, the most prominent terms get the biggest font. The simplest approach to tag cloud generation is using term frequency as measure for prominence: the more often a word occurs, the more important it is. In order to prevent highly frequent words without any content such as ‘the’ and ‘a’ to be very big, stop words are removed. This is the approach used by online tag clouds generators such as http://tagcrowd.com/.

In SWELL, we improve over this basic approach in two ways:

1. We do not only consider single words as terms, but also combinations of two or three words such as ‘knowledge worker’, ‘health care professional’ and ‘information retrieval’.

2. We do not only count how often a term occurs in a knowledge worker’s document set but also compare it to the term occurrences in a so-called background corpus. The purpose of the background corpus is to determine how specific/descriptive the term is for the personal corpus of the knowledge worker. For example, the word ‘do’ will occur in almost every document written in English, so although it might be frequent in the corpus of the knowledge worker, it is not very descriptive of the worker’s work domain. Terms that are much more frequent in the personal document collection than in general English, are the most descriptive of the personal document collection. The background corpus should be representative for general-use English, but could be more specific if we want to extract more specific terms. For example, in the artificial intelligence field, terms such as ‘data’ or ‘user’ are more frequent than in general English, but they might be considered too general to describe the work domain of one specific researcher in artificial intelligence.

   The corpus should be free to use, at least for research purposes, and not too big because it has to be distributed with the source code. We chose the Corpus of Contemporary American English as background corpus because the developers provide a word frequency list and n-gram frequency lists that are free to download.[1] A frequency list is much smaller in file size than a corpus (60MB in total) and easy to process.

We implemented three term scoring methods from the literature (see Section 2.2). We evaluated them on the document collections of five SWELL employees and, based on these evaluations, created an optimal combination of the three methods.

2.1 Pre-processing

The following pre-processing steps are performed on each document in the document collection.

1. Convert to plain text. The following formats are supported: txt, docx and pdf.
2. Split in sentences.
3. Extract candidate terms: Given a document collection, we consider as candidate terms all occurring n-grams (sequences of n words) that contain no stop words and no numbers. We use n=[1, 2,3] for the candidate terms. We save all candidate terms and their term counts in each corpus.
4. Extract term co-occurrences: For the foreground collection, we save all co-occurrence counts of terms within sentences. We need this information for one of the term extraction algorithms.

2.2 Term extraction

We implemented three different term scoring functions from the literature, aiming at giving the highest score to the most descriptive terms from the collection:

[1] For unigrams the 5000 most frequent words; for n-grams the 1,000,000 most frequent http://www.wordfrequency.info

1. Term frequency weighted with the frequency of the term in a background corpus using a parsimonious language model (Hiemstra et al. 2004). Parameter to be set: the strength of the contrast between foreground and background probabilities (lambda).
2. A term relevance score based on the distribution of co-occurrences of the term with frequent terms in the collection (Matsuo and Ishizuka 2004). The rationale is of this method is that no background corpus is needed, and that the most frequent terms from the foreground collection serve as background corpus. Parameter has to be set: the size of the set of most frequent terms (topfreq).
3. A term relevance score based on the expected loss between two language models (the foreground model and the background model), measured with point-wise Kullback-Leibler Divergence (Tomokiyo and Hurst 2003). Tomokiyo and Hurst propose to mix two models for term scoring: phraseness (how tight are the words in the sequence) and informativeness (how informative is the term for the foreground corpus). Parameter to be set: the weight of the informativeness score relative to the phraseness score (gamma).

The result of each of the term scoring methods is a list of terms for a document collection, with scores.

2.3 Evaluation and optimization

We asked SWELL colleagues to provide us with a collection of at least 20 documents that are representative for their work. This is the data that we obtained for five colleagues:

Colleague	# of documents	Indication of word count
Worker 1	29	280 K
Worker 2	21	350 K
Worker 3	23	260 K
Worker 4	13	260 K
Worker 5	26	300 K

For each of these document collections, we generated three lists with 300 terms each using the Hiemstra, Matsuo and Tomokiyo methods. Then we created a pool of around 150 terms per collection by linearly weighting the scores for the three methods and ranking the terms by the combined scores. These terms were judged in alphabetical order by the owner of the document collection. We asked them to indicate which of the terms are relevant for their work and provided instructions with some examples.

There was a large deviation in how many terms were judged as relevant by the users (between 24 and 51%), but generally, around one third of the generated terms was perceived as relevant.

	W1	W2	W3	W4	W5	All
# of terms judged	160	159	154	147	157	777
% of terms judged as relevant	49%	30%	29%	51%	24%	36%

The terms that were judged as non-relevant were classified in one of the following categories:

- too specific: term is very specific for (one) part of your work, e.g. 'correct answer', 'source authority sensitivity'
- too generic: term is not very descriptive because it is applicable to a lot of work, e.g. 'data', 'baseline', 'parser'
- incomplete: a partial term, e.g. ‘care professional’ instead of ‘health care professional’. Often, the complete term is also in the list.
- noise: can be words in a different language, a PDF conversion error, parts of the document structure, etc. E.g.: 'chapter', 'et al', 'conference', 'see section'
- not a term: words that you would not consider a descriptive term, e.g. 'using', 'words overlapping', 'new topic', 'million queries'
- other: this might be for example that you consider a term to be a correct term but it is not relevant enough for your work, or any other reason that is not one of the above.

With the term assessments, we evaluated the term extraction methods. We compared them to simple term frequency as scoring criterion. We chose the default value of 0.5 for the parameter lambda in the Hiemstra method. In the Tomokiyo method, we decided to give more weight to the phraseness component than to the informativeness component, because Tomokiyo is the only method that has a phraseness component. We set gamma to 0.1, which leads Tomokiyo to generate much more multi-word terms than the other methods. We should note here that the parameters should be optimized in future work.

As evaluation measure we used Average Precision, which evaluates a ranking by calculating precision at every position where a relevant term is found and then averaging over these precision scores (Zhu 2004).

	Average Precision					
	John	Maya	Rianne	Saskia	Suzan	All
TF scores	0.388	0.299	0.213	0.448	0.166	0.303
Hiemstra scores (lambda:0.5)	0.407	0.312	0.221	0.461	0.177	0.316
Matsuo scores (toprank:10)	0.424	0.319	0.217	0.441	0.207	0.322
Tomokiyo scores (gamma:0.1)	0.409	0.438	0.409	0.599	0.293	0.430

The table shows a large variation in the evaluation scores for the five knowledge workers. Of course, Average Precision is directly related to the number of terms that were judged as relevant. All three term extraction methods give better results than the plain TF scores and over all, the Tomokiyo method gives the best result.

We expected that each of the methods has its own strength, so we searched for an optimal combination of the three. To that end, we extracted the top 100 terms as calculated by each of the three methods. We first normalized each of the three scores relative to the minimum and maximum score in the term list:

$$score_{norm} = \frac{score - score_{min}}{score_{max} - \ score_{min}}$$

Then we combined the normalized scores assigned by the three methods linearly:

$$score_{combined} = \frac{w_H * score_H + w_M * score_M + \ w_T * score_T}{w_H + \ w_M + \ w_T}$$

Here, $score_H$ is the score assigned to the term by the Hiemstra algorithm, $score_M$ is the score assigned by the Matsuo algorithm and $score_T$ is the score assigned by the Tomokiyo algorithm.

To find the optimal combination of integer weights, we used a genetic algorithm (Mitchell 1996). We optimized the weights for the individual term lists (using Average Precision as criterion), and for all term lists together (using Mean Average Precision as criterion). The table below shows the optimal weights that we found:

	Optimal weights					
	W1	W2	W3	W4	W5	All
Hiemstra scores (lambda:0.5)	0	13	0	10	0	12
Matsuo scores (toprank:10)	49	65	1	0	2	2
Tomokiyo scores (gamma:0.5)	21	12	58	98	90	95
3 methods equally weighted	0.479	0.431	0.308	0.532	0.213	0.393
3 methods optimally weighted	0.539	0.488	0.408	0.611	0.294	0.441

Again, there is quite some variation in which method works the best for which document collection. This shows the importance of personalization in term-related SWELL@work components. On the other hand, we do want to avoid that the settings are too specific for an individual document set, since it limits the generalizability of the method. Therefore, these results are the starting point for further analysis and development.

For this deliverable, we implemented the optimal weight found for all knowledge workers together: (12,2,95) for (Hiemstra, Matsuo, Tomokiyo). This gives a Mean Average Precision of 0.441 on the training data.

2.4 Creating a SWELL tag cloud

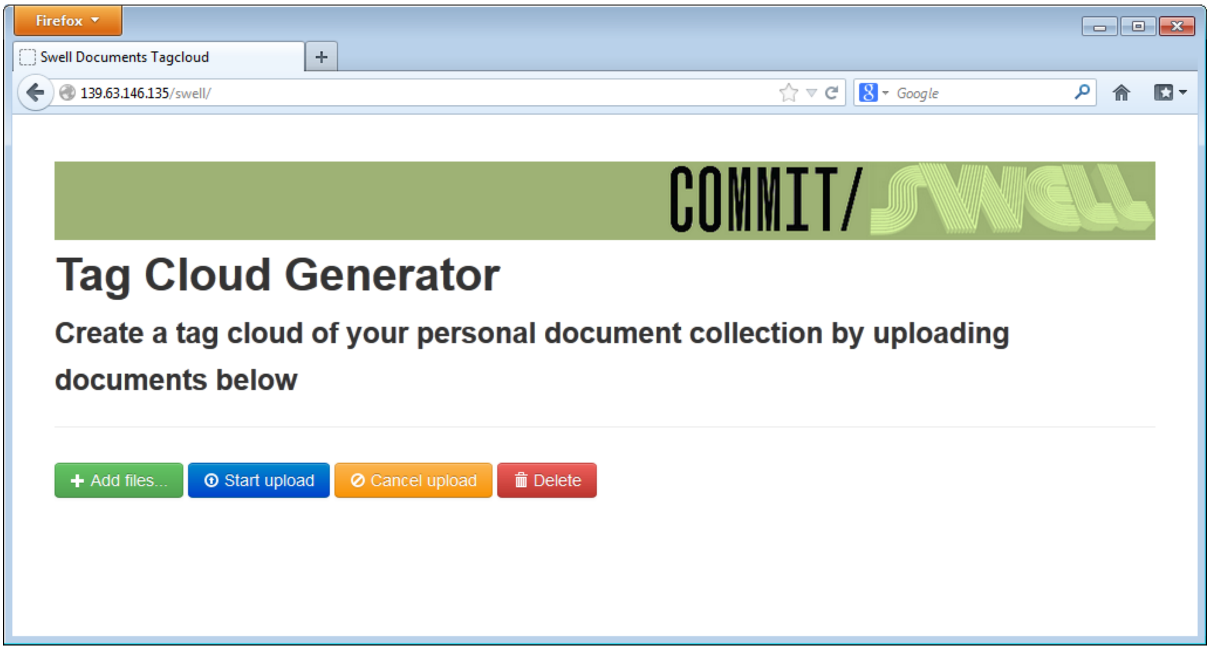
The term scores that are outputted by the optimal combination of the Hiemstra, Matsuo and Tomokiyo methods are used for generating a tag cloud: the higher the score for a term, the bigger its font size.

3 Graphical user interface

In the GUI that can be found on http://139.63.146.135/swell/, a user can upload a document set and is presented with two term clouds:

1. A tag cloud that is based on simple standard methodology, where the number of times a word occurs in the collection determines its font size.

2. A SWELL tag cloud, in which not only single words but also terms consisting of multiple words (e.g. ‘knowledge worker’) are shown, and a more sophisticated term scoring is applied: the optimal combination of term extraction methods found in our experiments.



Tag Cloud Generator

Create a tag cloud of your personal document collection by uploading documents below

4 Future developments

- In modelling the user, we plan to develop the idea that each project or topic that a user works on has its own terminology and social structure (collaborators). That means that a user model will be facetted in partial models. We will experiment with the implementation of these facetted models by automatically clustering the user’s documents

- We will further improve the term scoring algorithms by (1) experimenting with a different background corpus that is more related to the work domain of the user (now, terms such as ‘data’, ‘model’ and even ‘et al’ are ranked high as prominent terms); (2) optimisation of the parameters for the term extraction methods; (3) clear out incomplete terms such as ‘care professional’ when ‘health care professional’ is also in the list.

- We aim to evaluate the term extraction technology in the context of two tasks: (1) Prioritizing e-mail messages based on the user model of the receiver and (2) Adapting search results based on the user model (results that match the user’s interest are ranked higher). For both tasks we have started our work and reported preliminary results in two papers (Sappelli et al. 2013; Verberne et al. 2013).

- One of the other possible future applications is to use terms as descriptors for annotating a work-life log. It would be interesting to have the functionality to navigate to related topics using a taxonomy of terms. We will start explorations in this direction soon.

5 Literature

- D. Hiemstra, S. E. Robertson, and H. Zaragoza (2004) "Parsimonious language models for information retrieval". In *Proc. 27th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 178–185, Sheffield, United Kingdom.
- Y. Matsuo and M. Ishizuka (2004). "Keyword extraction from a single document using word co-occurrence statistical information", *International Journal on Artificial Intelligence Tools*, Vol. 13, No. 1, 157-169.
- M. Mitchell (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press. ISBN 9780585030944.
- T. Tomokiyo and M. Hurst (2003). "A language model approach to keyphrase extraction". In *Proceedings of the ACL Workshop on Multiword Expressions*.
- S. Verberne, M. Sappelli, D. R. Sørensen, W. Kraaij (2013). Personalization in Professional Academic Search. In: M. Lupu, M. Salampasis, N. Fuhr, A. Hanbury, B. Larsen, H. Strindberg (eds.): Proceedings of the Integrating IR technologies for Professional Search Workshop, Moscow, Russia, 24 March 2013
- M. Zhu. "Recall, precision and average precision." Department of Statistics and Actuarial Science, University of Waterloo, Waterloo (2004).
- M. Sappelli, S. Verberne, W. Kraaij (2013). Combining textual and non-textual features for e-mail importance estimation. Submitted to SIGIR.