Computers in Industry 78 (2016) 16–28

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A distributional approach to open questions in market research

Stefan Evert [a], Paul Greiner [a,*], João Filipe Baigger [b], Bastian Lang [b]

[a] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Department Germanistik und Komparatistik, Professur für Korpuslinguistik (Corpus Linguistics Group), Bismarckstr. 6, 91054 Erlangen, Germany
[b] Rogator AG, Emmericher Str. 17, 90411 Nürnberg, Germany

ARTICLE INFO

Article history:
Received 13 December 2014
Received in revised form 8 October 2015
Accepted 13 October 2015
Available online 29 November 2015

Keywords:
Topic clustering
Distributional semantics
Sentiment analysis
Market research

ABSTRACT

Free-text responses to open questions are a rich and valuable resource in modern-day market research, but often pose problems for a traditional analysis, which requires prohibitively expensive manual coding of topic categories. The Klugator Engine (TKE) is a system for semi-automatic identification, exploration and visualization of topics and sentiment in large collections of such free-text responses or other short text fragments. The system utilizes state-of-the-art techniques of natural language processing and machine learning to transform textual input into a structured corpus, complemented by automatically determined polarity scores for individual responses. Statistical and distributional methods are then applied in order to identify semantic clusters of responses, label each topic cluster with a set of salient keywords, and evaluate the sentiment associated with the topic. This process can run in fully automated fashion, but it also offers the opportunity of interactive parameter tuning and refinement guided by the end user. Results are presented in a concise graphical visualization supported by detailed tables with numerical information. Embedded in RogTCS, the Rogator Text Clustering Solution, TKE enables customers to obtain a good overview of the main topics in a text collection comprising thousands of responses within 20 min of interactive exploration. An evaluation study based on a data set of more than 60,000 word tokens has shown good agreement with the topics identified by manual coding, rendering TKE a powerful tool for the analysis of unstructured textual data.

1. Overview

This article describes the purpose, design and implementation of The Klugator Engine (TKE), a specialized text clustering software for online surveys developed by the Corpus Linguistics Group at Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) in co-operation with local market research company Rogator AG. TKE has already been applied successfully to the analysis of data collected in online surveys, but it could also be used for a variety of other tasks on similar kinds of text data.

1.1. Motivation

A large proportion of the data collected in market research are strictly quantitative in nature, usually obtained from opinion polls with predefined answer options. While such data are easy to analyze with standard techniques of descriptive and inferential statistics, there are two important drawbacks: (i) usually the level of measurement is restricted to categorical and numerical scales, which cannot account for qualitative aspects and complex relationships; (ii) they do not adequately reflect the true variety of respondents' opinions, discarding any information that the researcher designing a poll did not have in mind a priori.

Open questions permitting unrestricted free-text answers give much more fine-grained insights, but they are often shunned as an alternative (or complement) to quantitative polls because their analysis is time-consuming and expensive. All responses have to be examined manually and assigned to topic categories, based on an ad-hoc code plan specially designed for each question. In this way, the qualitative data are transferred into a quantitative representation, which can then be analyzed with the usual statistical techniques. Code plans can rarely be reused without discarding important aspects of the unrestricted answers, and they often need to be revised and extended during the coding process.

With the recent trend towards online polls and other electronic forms of surveys, large amounts of free-text responses can easily be collected in machine-readable form. It is not uncommon for an online poll to produce several tens of thousands of responses. Open

* Corresponding author.
E-mail addresses: stefan.evert@fau.de (S. Evert), paul.greiner@fau.de (P. Greiner), f.baigger@rogator.de (J.F. Baigger), b.lang@rogator.de (B. Lang).

http://dx.doi.org/10.1016/j.compind.2015.10.008
0166-3615/

questions are sometimes included merely to improve the user experience, and are then discarded when analyzing the survey.

TKE was developed in order to enable market researchers to exploit this wealth of qualitative information. It is able to carry out a mostly automatic identification of key topics within a large set of unstructured free-text answers to open questions, to determine the general sentiment expressed towards each topic, and to visualize the quantitative distribution of topics among the answers. The software is already in commercial use as the core engine of the Rogator Text Clustering Solution (RogTCS), distributed by German market research company Rogator AG.

1.2. Goals

Key requirements for TKE include (i) fast automatic analysis (so that the open questions of a typical poll can be explored interactively by customers), (ii) ease of use (as RogTCS is operated directly by customers), (iii) domain independence (so that all open questions can be analyzed) and (iv) support for multiple languages (for use with international surveys). Therefore, the system has been designed to use as few language- and domain-specific resources as possible, which also helps to avoid steep licensing costs for ontologies, sentiment lexica, etc. Instead, the main text clustering component of TKE builds on unsupervised statistical procedures that have been carefully tuned in order to give a faithful representation of the content of the input data, condensing the main topics and sentiments into a manageable amount of graphs, numerical indicators and tables.

In most cases, TKE is able to generate satisfactory results without any manual intervention, so that a rough analysis of a text collection can be carried out within a few seconds. If desired, this initial analysis can afterwards be refined by users in an interactive, semi-automatic procedure.

1.3. Scope of application

TKE is commercially available, providing the core engine of RogTCS, a tool for fast and cost-effective analysis of open questions in market research.[1] Therefore, the typical text material processed by TKE at this point consists of short textual answers and comments from online surveys concerned with a range of different products and services. Texts may be in different languages, with a main focus on German and English. RogTCS offers market researchers an alternative to the time-consuming and expensive manual coding of answers to open questions, which involves the creation and application of a specialized code plan for each question.

While optimized for online surveys, TKE can also be applied to other kinds of digital collections of natural language data, provided that they consist of short, focused texts and contain enough material for a statistical topic analysis. Possible use cases are, among others, messages (tweets) sent via Twitter or similar micro-blogging services, online discussion boards, customer product reviews, as well as other material obtained from various social media platforms.

1.4. An example session

In order to give readers a better idea of the functionality of TKE and the quality of its results, this section presents an example analysis based on a real-life data set. The data comprise 4627 responses to an open question posed in an online survey about the redesign of an e-mail provider's Web site, written in English. The average length of a response in this data set is 13.7 words, resulting in a total size of 63,314 word tokens. The data were analyzed with the most recent TKE version (as of December 2014), coupled with a Web-based GUI that has also been used for development and in-house testing of the engine (see Section 3.4 for more information).

After uploading the text data, TKE carries out a fully automated analysis and presents its results in the form of a semantic map, as shown in Fig. 1. The quality of this initial analysis depends strongly on the quality of the input data – the statistical topic identification works best with focused responses to a clearly phrased question – and might not always be as neat and easily interpretable as the example presented here. A key parameter is the number of topic clusters, which has to be specified by the user in the current TKE implementation. The result in Fig. 1 was achieved after a few minutes of manual experimentation by reducing the number of clusters from its default value of 20. The interactive development GUI, which is similar to the RogTCS user interface, allows users to explore different settings for a wide range of system parameters with ease. Note, however, that only the number of clusters – a simple and intuitive parameter – had to be adjusted in this example.

Each circle in Fig. 1 represents one of the automatically identified topic clusters, labelled with single words or bigrams (pairs of consecutive words) that occur frequently in responses assigned to the corresponding cluster. While labels are not perfect, it is usually easy for human users to infer the main topic of a cluster. For example, the labels *like*, *new*, *service* clearly indicate that respondents enjoyed the new product. Multiword expressions consisting of more than two words (e.g. *good spam filter*) are broken down into multiple labels (*good spam*, *spam filter*) that can easily be pieced together by users. If necessary, users can request additional label suggestions or directly inspect the answers assigned to a topic cluster.

The area of each circle reflects the number of responses assigned to the topic, which we refer to as the *mass* of the topic. The colours of the circles indicate the overall sentiment expressed towards topics, using a colour palette ranging from green (for positive sentiment) to red (for negative sentiment). Neutral or mixed sentiment corresponds to a yellow hue. The grey circle in the centre stands for responses that could not be assigned reliably to one of the topic clusters, e.g. because they are formulated in an unusual way, express a very infrequent opinion, or consist of meaningless text. In this case, the data set contains 3698 well-formed responses (excluding e.g. empty answers), out of which 3156 were automatically assigned to one or more topic clusters by the engine.

Fig. 2 gives an alternative overview of the main topics in tabular format, including detailed quantitative information. Market researchers are particularly interested in the importance of each topic as represented by its mass, i.e. the number of responses in which the topic was addressed. Notice that these counts do not add up to the total number of texts because TKE can assign a response to multiple topic clusters. Here, 829 responses were (at least partially) assigned to the most frequent topic (C11: *ease of use*); 1028 responses contain a text fragment that does not fit any of the topics and are thus assigned to the residual cluster R01 shown as a grey circle in the map; and 1193 responses are either empty or include a fragment that cannot be analyzed by the system at all (because all words have been filtered out by the stopword list and frequency threshold, cf. Section 3.2). Additional columns indicate the quality of each topic cluster (based on the homogeneity of the corresponding text fragments) and the average sentiment towards the topic (cf. Section 3.1).

Some of the responses from topic clusters C09: *web interface, good interface, user interface* and C06: *time, problems, signed* are shown in Figs. 3 and 4. The responses are ranked by their

[1] https://www.rogator.de/software/textanalysesoftware.html.

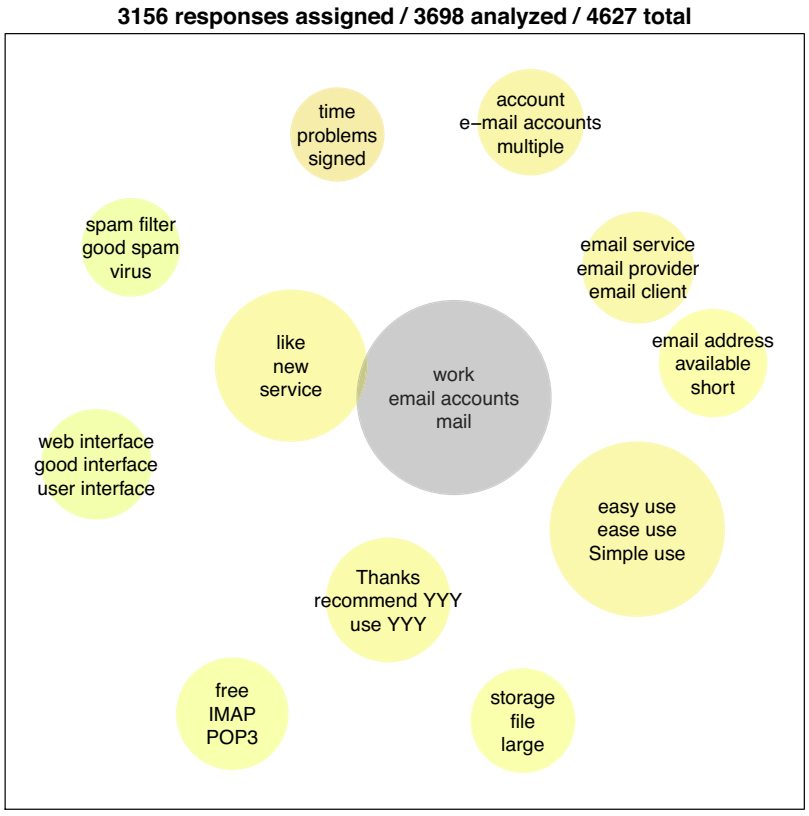

Fig. 1. Semantic map visualizing the main topics mentioned in the responses to an open question from an online survey about the redesign of the Web site of an e-mail provider.

prototypicality for the topic, i.e. how similar they are to the cluster prototype. These screenshots also show that many responses are assigned to multiple topic clusters. Parts of a response that are not relevant for the current topic are automatically greyed out.

A quick look at Figs. 3 and 4 confirms and clarifies the straightforward interpretation suggested by the automatically generated labels. On the one hand, C09 (327 responses) shows that respondents seem to be quite happy with the new design of the Web site in terms of navigation and general attractiveness (*Slick interface* ...). On the other hand, C06 (237 responses) indicates that a considerable number of users are experiencing issues with the responsiveness of the new page (*It takes too long...*).

	mass	type	labels	quality	density	sentiment
C11	829.0	normal	easy use, ease use, Simple use	0.41	59.4	0.02
C03	624.0	normal	like, new, service	0.22	77.8	0.04
C05	416.0	normal	Thanks, recommend YYY, use YYY	0.45	55.3	0.05
C04	339.0	normal	free, IMAP, POP3	0.49	54.2	0.13
C07	334.0	normal	email service, email provider, email client	0.44	52.4	0.02
C09	327.0	normal	web interface, good interface, user interface	0.43	59.7	0.19
C02	314.0	normal	email address, available, short	0.36	60.1	0.07
C01	303.0	normal	account, e-mail accounts, multiple	0.36	59.0	0.02
C08	293.0	normal	storage, file, large	0.36	64.8	0.11
C10	260.0	normal	spam filter, good spam, virus	0.45	57.4	0.19
C06	237.0	normal	time, problems, signed	0.35	64.4	-0.04
R01	1028.0	rest	work, email accounts, mail	0.02	83.9	0.12
OUT	1193.0	out	unanalysed documents	0.00		-0.01

Fig. 2. A tabular overview of the identified topics with quantitative information.

2. Comparison with other work

2.1. Related commercial systems

In the field of market research, several computer-aided systems are available for the extraction of topics, sentiment and other information from unstructured text. For example, IBM, SAP and SAS offer solutions for text analysis within their general analytics products, while companies like Attensity and Lexalytics specialize in this field. We are not in a position to evaluate these systems in detail, but according to a survey conducted by Rogator, most of them use a deductive and rule-based approach, i.e. they rely on pre-existing categories (in the form of coding schemes), extensive lists of keywords for topic detection and sentiment analysis, as well as comprehensive taxonomies and ontologies.

While such approaches often achieve high classification accuracy, there are several drawbacks. Manual (or semi-automatic) compilation of the required knowledge bases and resources is time-consuming and expensive, especially if the system has to be adapted to a new domain or language. Only data from supported domains, question types and languages can be analyzed immediately. Otherwise, the necessary adaptations to the system (new coding schemes, keyword lists, ontologies) will take several hours or days and will rarely be profitable for one-off surveys. Adaptation to a new language is even more costly and will only be amortized over the course of a large number of surveys. Finally, such a system can only identify topics it already "knows about", limiting the breadth and completeness of results in a manner similar to the sets of predefined answers in traditional quantitative polls. According to our industry partner, preliminary tests with several commercial rule-based systems showed that they often leave a large portion of the responses unassigned, at least unless customers manually

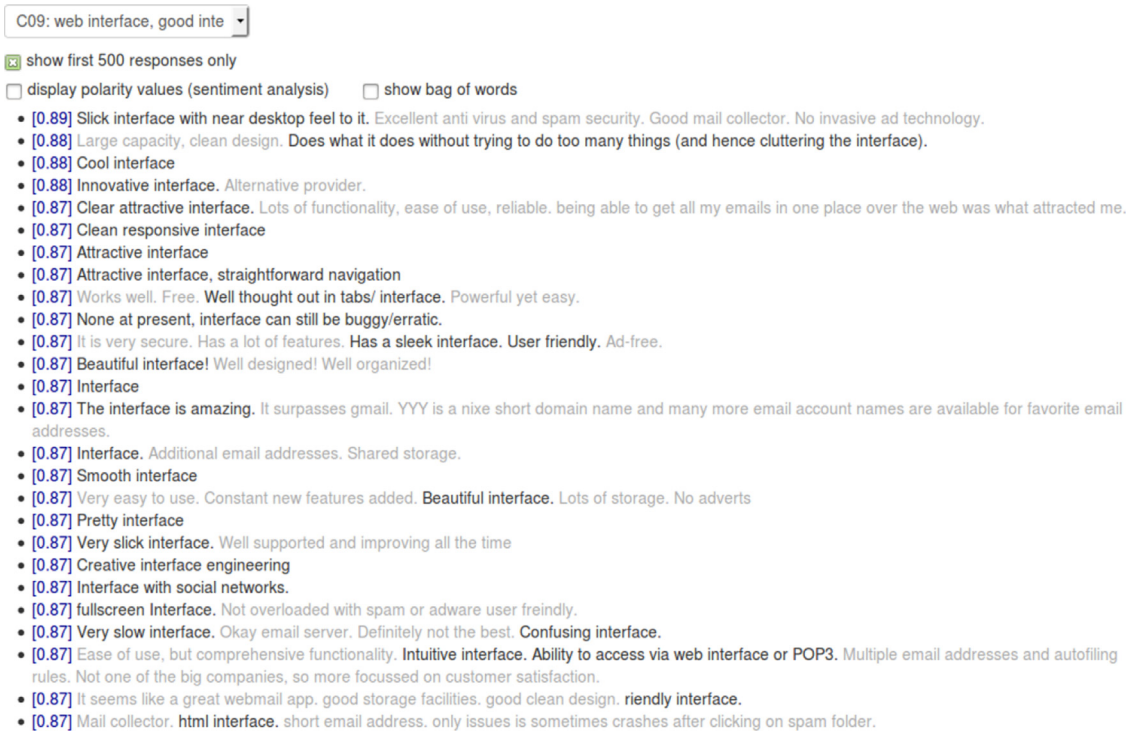

Fig. 3. Responses assigned to topic cluster C09 with labels *web interface*, *good interface* and *user interface*.

create a large domain-specific ontology or purchase corresponding modules.

By contrast, TKE was designed to work in an inductive manner, for the most part relying only on the information and statistical patterns contained in the input texts. We found that even without external knowledge resources, surprisingly accurate results can be obtained if a sufficient amount of input data are available (approx. 1000 responses) and if the parameters of the statistical techniques are carefully tuned. This enables a completely domain-independent analysis that is neither limited to particular surveys and questions nor prejudiced by predefined coding schemes, ontologies, etc. Domain-independence does not automatically imply language-independence, of course. Some parts of TKE rely on language-specific resources for a linguistic analysis of the input texts: manually labelled training data for the sentiment classifier, a stemming algorithm, and a list of stopwords. If such resources are

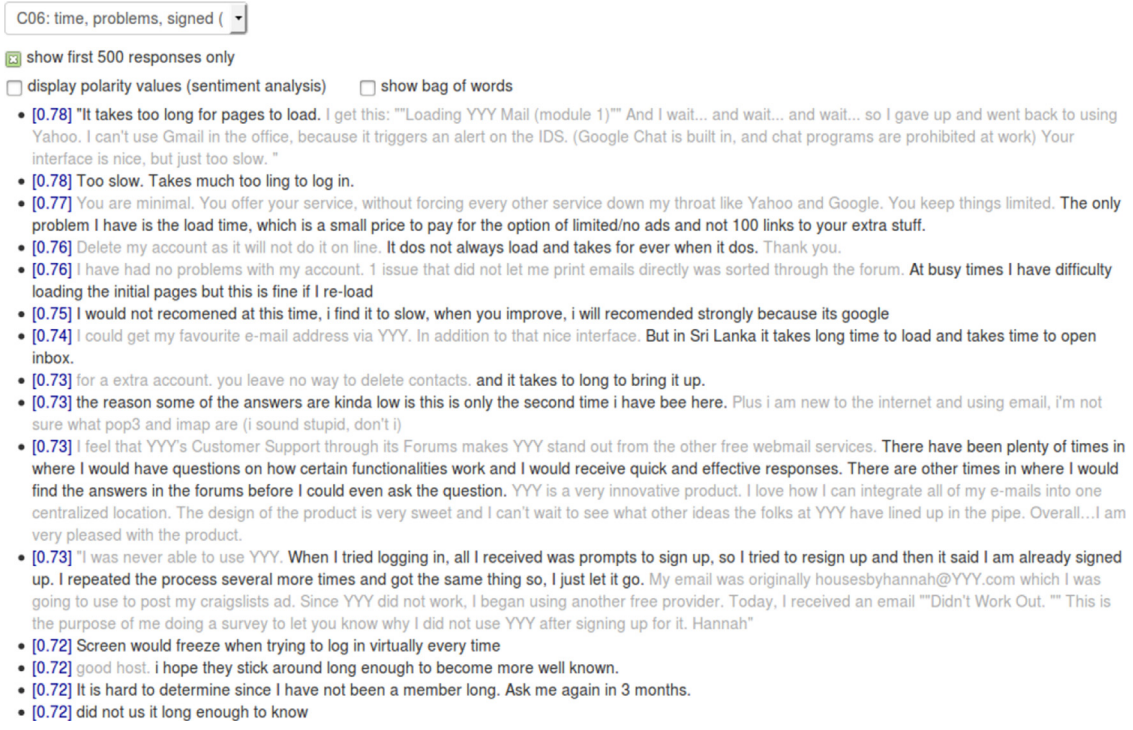

Fig. 4. Responses assigned to topic cluster C06 with labels *time*, *problems* and *signed*.

not yet available for a new language, TKE can also operate in a basic mode, where the sentiment analysis module is disabled and only a minimal linguistic analysis is carried out. This ensures that the engine can immediately be used with data in any language, as long as a suitable stopword list can be procured.

2.2. Related research

Since automated text analysis nowadays has a wide range of possible applications, the field has also become of significant interest to academic research. For text clustering, the main purpose of TKE, Zhang et al. [1] give a comprehensive overview. A major difference between TKE and some of the proposed methods is the utilization of single words and bigrams in a bag-of-words representation, ignoring word order, phrase structure, and larger chunks of text such as named entities or complex technical terms. TKE also uses a flat clustering technique instead of hierarchical clustering. In other respects, there are many similarities: the application of frequency thresholds, the possibility to accept unassigned material into a predefined cluster and the general preprocessing of input texts. Since one of the main requirements for TKE was to be robust, fast and not too memory-hungry, we decided to rely on tried and tested methods that can be implemented efficiently (cf. Section 3). The next paragraphs discuss existing approaches more thoroughly and compare them with TKE.

Among the most successful approaches to topic analysis and text clustering are probabilistic topic models such as latent Dirichlet allocation (LDA, [2]). Preliminary experiments during the design phase of TKE showed poor clustering quality on our data sets, though. We believe that this is connected to the short and focused nature of our documents. While the average length of a response is 13.7 words in the example data set from Section 1.4, it is reduced to only 7.7 words by the stopword filter and sentence splitting algorithm. Many responses only consist of one or two relevant content words (e.g. *storage* or *spam filter*). LDA was designed for larger documents that contain a mixture of topics rather than such concise responses addressing a single topic. Our tests of LDA also showed a strong tendency for frequent words to be assigned high probability in many of the latent topics (e.g. *account* or *easy* in the example data set). The initial term clustering carried out by our algorithm (cf. Section 3.2) ensures that most words are associated with a single topic, which is often appropriate for survey responses.

Li and Chung [3] propose an algorithm called clustering based on frequent word sequences (CFWS) that focuses on documents as sequences of words for computing semantic similarity. This focus on sequences poses a major difference to the system presented in this article. Even though it achieves impressive results in other areas, this approach is unsuitable for the type of data TKE was designed to process for the simple reason that sequences of more than two words rarely occur with high frequency in survey responses. Another difference is that CFWS automatically determines the optimal number of clusters, a feature currently not supported by TKE. The reason for this lies in the fact that, according to experiments of Rogator, interactive experimentation with different numbers of clusters improves the user experience, allowing customers to explore the data set in a playful manner rather than being constrained to a single analysis.

Another approach is presented by Beil et al. [4] with frequent term-based text clustering (FTC) and hierarchical frequent term-based clustering (HFTC). FTC produces a flat clustering while HFTC exploits so-called lattice structures to build a hierarchical clustering. To produce these lattice structures, *n-term sets* are identified within documents, where *n* corresponds to the depth of the hierarchy. As has been mentioned above, TKE uses flat rather than hierarchical clustering because flat clusters are much easier to manage and update as users explore different parameter settings interactively. Moreover, we do not expect to find many recurrent *n*-term sets in the typical input data of TKE. Similarities can be found in the rejection of low-dimensional frequent term sets for semantic analysis by FTC and HFTC, which corresponds roughly to the frequency threshold utilized by TKE (cf. Section 3.2). Beil et al. [4] also agree with us that k-means is – despite the availability of much more sophisticated methods – a robust clustering tool in the sense that it combines remarkable cluster quality with high efficiency.

The application of hierarchical clustering is also a major difference between our system and the approach of frequent itemset-based hierarchical clustering ([5], FIHC), which is based on frequent sets of words occurring together in a certain minimum proportion of the documents in a cluster. Our reasons for not implementing this approach are similar to the arguments put forward in the comparison with HFTC above.

Another line of related academic work is ontology construction, which aims at "capturing knowledge about the world explicitly for a specific task" [6, p. 577]. Ontology construction starts out with the extraction of terms based on a morphological and syntactical analysis of the input documents, followed by the identification of synonyms, usually relying on information from external resources such as WordNet. Even though state-of-the-art methods for ontology construction show good results and would complement TKE's functionality, such approaches were deliberately avoided because of their reliance on resources that are either expensive or not licensed for commercial use.

Fields like expertise mining and expert finding [7] make use of related techniques. Expertise mining aims to identify skills and competencies and, in this approach, is based on manually identified domain-specific terms inserted into common textual patterns with an ensuing filtering process based on external Web search engines. For expert finding, these results are combined with information about the authors of, or people with access to processed documents. Even though these are interesting ways of assessing the contents of textual documents, TKE is designed to operate on material reflecting the general knowledge and subjective opinions of laypeople rather than highly specialized expert skills.

The most similar software solution in the academic realm we are aware of is the topic analysis module of OntoGen [8], an integrated toolkit for the semi-automatic construction of domain ontologies. Like TKE, OntoGen uses a bag-of-words representation of documents in a vector space model and applies k-means clustering. Despite such similarities, the tools differ considerably in their main focus: OntoGen builds hierarchies of clusters and concepts in order to support ontology construction, whereas TKE has been optimized for the computer-assisted analysis of survey responses and other short, focused texts.

3. The TKE system

TKE is designed as a pipeline of three separate modules corresponding to the three phases of a typical analysis (Fig. 5). When a user has uploaded a new data set, the preprocessing module carries out a linguistic analysis of the texts, determines negative and positive sentiment, and collects metadata information included in the data set (such as gender and age of respondents or different focus groups). The core module of TKE then carries out a fully automated topic analysis, generates labels for the topic clusters, and determines the distribution of sentiment polarity and metadata categories across clusters. This module can be re-run with different parameter settings, allowing experienced users to fine-tune the analysis results. Finally, an interactive refinement of the topic clustering can be achieved by repeated

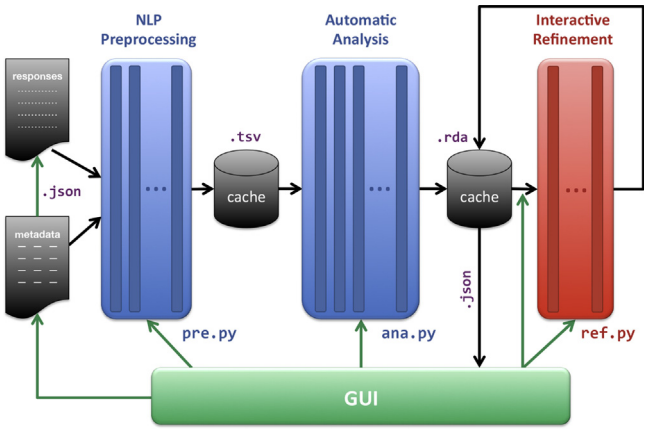

Fig. 5. Overview of the modularized architecture of TKE. (For interpretation of the references to color in text near the reference citation, the reader is referred to the web version of this article.)

application of the refinement module, shown in red in Fig. 5. The entire pipeline is managed by the external GUI.

Advantages of the modularized implementation are its maintainability and easy extensibility, as well as the possibility of replacing individual components with improved alternatives. It also enables the external GUI component to repeat individual steps of the analysis without the unnecessary overhead of restarting the entire pipeline and preprocessing phase. This ensures a highly interactive user experience with a responsive GUI, especially during manual tuning of the analysis parameters and during the refinement phase. For this purpose, the output of each module is serialized to a cache file that serves as input for the next stage of the pipeline. Cache files are small enough to be retained throughout a user session and serve as an unlimited "undo" history.

The following three sections describe the approaches and methods used by each of the three modules: preprocessing in Section 3.1, automatic topic analysis in Section 3.2, and interactive refinement in Section 3.3. Section 3.4 provides some information about our implementation of the TKE system.

3.1. Preprocessing

The purpose of this module is to convert unstructured plain-text input into a numerically indexed corpus format that can be processed efficiently by the subsequent modules. It uses various standard techniques of natural language processing, which have been optimized for the requirements of TKE. We consider sentiment detection to be a part of the preprocessing that is applied, if desired, before the more time-consuming and elaborate topic clustering algorithm.

In the first step, texts are split into sentence-like units, which are then further divided into single word tokens. Both tokenizers are based on implementations in NLTK [9] and have been specially adapted to the development data provided by Rogator. The main goal of the modifications was to pay respect to common non-words like smileys and URLs, which standard tokenizers tend to break apart. TKE circumvents this behaviour by applying transformations based on regular expressions. The resulting tokens are normalized by case-folding and application of the Snowball stemming algorithm [10]. Stemming was chosen over lemmatization since it is far less dependent on lexical resources. Stemmers are available for a wide range of languages; if necessary, a stemmer for a new language can often be implemented with relative ease. The normalized tokens are filtered against a stopword list that includes all function words and has gradually been extended during the development and testing process by Rogator. The default stopword lists are fairly small and domain-independent, but can be customized by users with project-specific additions.

The second step is an optional sentiment analysis at sentence level, producing a polarity score (ranging from −1 = highly negative to +1 = highly positive) and a polarity category (negative, neutral, positive) for each sentence. TKE's sentiment analysis component is based on SentiKLUE [11,12], a machine-learning polarity classifier that has achieved state-of-the-art performance in recent SemEval competitions. Since SentiKLUE makes use of polarity lexicons and other resources that are only licensed for academic use, a simplified version of the system was implemented and trained on in-domain text samples annotated by Rogator. The simplified system applies a Maximum Entropy classifier to word frequencies and the occurrence of positive and negative emoticons as features. For each document, the posterior probabilities of the categories "positive" (p_+), "neutral" (p_0) and "negative" (p_-) are combined into a single sentiment score by taking the difference $p_+ - p_-$, which ranges from −1 (certainly negative) to +1 (certainly positive). The label "positive" is assigned only if $p_+ > 0.6$, the label "negative" only if $p_- > 0.6$; otherwise the sentence is considered to have neutral sentiment (see Section 4.1 for the rationale behind this additional bias).

Approximately 10,000 sentences of training data were available for each supported language (German and English). Comparative experiments showed that this simple classifier performs better than the full SentiKLUE system trained on out-of-domain data. A quantitative evaluation of the German polarity classifier is reported in Section 4.1. Since this step relies on language-dependent models for the classifier, it is currently skipped for languages other than German and English, allowing TKE to operate in a basic mode across a wide range of languages. If desired, the sentiment analysis can be extended to further languages with reasonable expense (i.e. manual classification of 10,000 sentences by non-expert annotators).

Finally, each sentence is transformed into a bag-of-words representation. In addition to the individual word tokens, frequent bigrams (pairs of consecutive words) are identified and included as separate units in the bag of words. We extract bigrams according to the following criteria: In a first step, all possible bigrams are determined. The resulting list is filtered against the stopword list and elements that contain a stopword in either position are discarded. TKE also stores the unprocessed input text and links it to the corresponding bag-of-words representations, so that the system output can display responses in their original form.

Following market research terminology, we refer to the input texts as *responses*. TKE implements multiple cluster assignment – which is often necessary in topic analysis, as the examples in Section 1.4 have shown – by splitting responses into sentence-like fragments that are called *documents* because they play a role similar to documents in vector-based information retrieval. Each document is assigned to a single topic cluster, but most responses consist of multiple sentence fragments that can be assigned to different clusters, resulting in a multiple assignment for the entire response.

3.2. Automatic topic analysis

This module applies statistical and distributional techniques to the preprocessed corpus in order to identify key topics in the form of document clusters. It also computes the polarity and strength of the sentiment towards each topic as well as its distribution across metadata categories. The topic clustering algorithm of TKE is controlled by a large number of parameters, whose default values have been carefully tuned in order to produce a good initial analysis for most data sets. No attempt is made to adapt parameters (such as the number of topic clusters) automatically

to a particular data set. Instead, end users can adjust the parameter settings in an interactive exploration phase, allowing them to guide the system towards a meaningful and interpretable analysis. Early feedback from Rogator's customers confirms that users enjoy such an interactive design and appreciate having a measure of control over the system. In what follows, we describe the most important parameters of TKE and their default settings.

Our basic approach combines the vector space model of information retrieval [13] with ideas from distributional semantics [14]. In a first step, a term-document matrix is generated from the bag-of-words representations of input documents. The rows of this matrix can be used to determine the semantic similarity of words and bigrams based on their distribution across documents. The columns are bag-of-words vectors that can be used to identify similar documents and group them into topic clusters. For both purposes, TKE uses angular distance between vectors as a metric (which is equivalent to cosine similarity). Dimensionality reduction by truncated singular value decomposition (SVD, see [15]) to 35 latent dimensions smoothes the highly sparse term-document matrix and allows the system to exploit higher-order information while discarding noise. In contrast to probabilistic topic models such as LDA, we do not assume that the SVD dimensions can be interpreted as topics. The main purpose of dimensionality reduction in TKE is to improve the term and document similarity information encoded by the co-occurrence matrix.

In a second step, a term clustering algorithm is applied to the rows of the SVD-reduced matrix, producing clusters of words and bigrams that provide an initial inventory of topics. TKE uses k-medoids clustering [16, chapter 2] – based on cosine similarity and with k set to the desired number of topic clusters – for this purpose, which we found to produce good and dependable results (unlike e.g. hierarchical clustering with Ward's method).[2] Note that this component only makes use of information derived from the originally supplied text data. No ontologies or other expensive resources are used, and the system does not need to be adapted to different domains. In order to achieve good clustering quality without external semantic knowledge, we found that it is essential to filter the words and bigrams with a finely tuned frequency threshold, which depends on the size of the data set. The current default heuristic $f \geq n^{0.8}/50 + 1$, where n is the number of documents (i.e. sentence-like units), works surprisingly well for most data sets.[3]

In a third step, the prototype vectors of the term clusters are used to initialize a document clustering algorithm, exploiting the fact that SVD maps both row and column vectors to a common latent space. The prototypes are updated iteratively by a variant of k-means clustering [17, pp. 424–428], which assigns each document to the nearest prototype and then recomputes prototypes from the set of cluster members. Our truncated version of k-means discards any documents for which no clear assignment is possible – i.e. those close to the boundary between two clusters – in order to create homogeneous topic clusters. While this is still a hard clustering algorithm, the individual documents (i.e. sentences) of each original response may be assigned to different clusters, so that the final output allows for soft or multiple assignment of responses. Documents that are discarded by the truncated k-means algorithm are collected in a separate residual cluster for manual inspection.

Finally, we use keyword analysis, a technique from corpus linguistics, to generate meaningful labels for the topic clusters. Keywords are words and bigrams that occur frequently in the topic cluster at hand, but are relatively infrequent in the remaining documents [18]. In TKE, any words or bigrams that have not been filtered out by the frequency threshold are considered as cluster labels. They are ranked according to the test statistic of a likelihood-ratio test, which is widely used in computational linguistics for this purpose [19], and the most relevant ones are displayed to the user. The topics are usually visualized in the form of a semantic map as shown in Fig. 1. For the construction of such a map, each cluster is represented by its prototype within the vector space model, which is then projected into a two-dimensional coordinate system using non-metric multidimensional scaling [20, pp. 306–309], so that distances between the different prototypes are approximately preserved. In this way, topic clusters with similar meaning will be positioned near each other in the semantic map.

The computation of frequency distributions for polarity and metadata categories across the clusters is straightforward, but care has to be taken to obtain correct counts at the level of responses. With multiple assignment, a single response may add to the mass of several topic clusters – this is appropriate because mass represents the number of respondents that mention a certain topic. On the other hand, responses must not be counted twice for the same topic, even if they are split into several sentences.

Advanced users can control many parameters of the automatic analysis in order to improve the clustering solution. The most important parameters are the number of clusters (with a global default value of 20) and the clustering margin, which determines how many documents are weeded out by the truncated k-means algorithm. The latter offers users a trade-off between thematic uniformity of clusters and the amount of material that can be analysed automatically. It is based on the ratio of distances between a document and two competing cluster prototypes, and is set to a default value of 5% (i.e. documents are discarded if the two distances are within 5% of each other). Other useful parameters for more advanced users are the frequency threshold for terms and bigrams (which may need to be lowered from its heuristic default for large data sets) and the number of latent SVD dimensions (to control the degree of smoothing applied to the term-document matrix).

In the latest version of TKE, users can also define additional stopwords in the analysis phase, or provide lists of synonymous expressions that are specific to a particular data set. For example, the topic labels in Fig. 1 could be simplified by declaring the expressions *easy use*, *ease (of) use* and *Simple use* to be synonymous, so only one of them would be shown as a representative label. These parameters operate directly on the term-document matrix, thus avoiding the overhead of re-running the preprocessing module.

3.3. Interactive refinement

After the fully automatic analysis, users have the option of further improving the topic clusters through an interactive refinement procedure. In each step, the user selects one of two operations illustrated in Fig. 6.

The SPLIT operation divides a cluster into a specified number of sub-clusters, introducing additional prototypes. This step is useful

[2] For example, in the analysis described in Section 1.4, the term clustering grouped the words *attachments, capacity, file, handling, large, send, size, storage* as well as the bigrams *attachment size, large attachments, large files, large storage, send large, storage capacity* together. This example shows that it would make little sense to evaluate TKE's term clustering against a general semantic network or ontology: while *large* and *send* have an entirely different meaning than words such as *attachment* or *storage*, they were predominantly used in the context of sending e-mails with large file attachments in this particular data set, so it is in fact appropriate to include them in the cluster.

[3] The heuristic was developed by manual experimentation during the development and testing of TKE. Some concrete examples might give a better idea of the typical frequency thresholds used: for a data set of 1000 documents (the smallest size for which TKE is designed), the threshold is $f \geq 6$; for a data set of 5000 documents it is $f \geq 19$; for a very large data set of 20,000 documents it is $f \geq 56$.

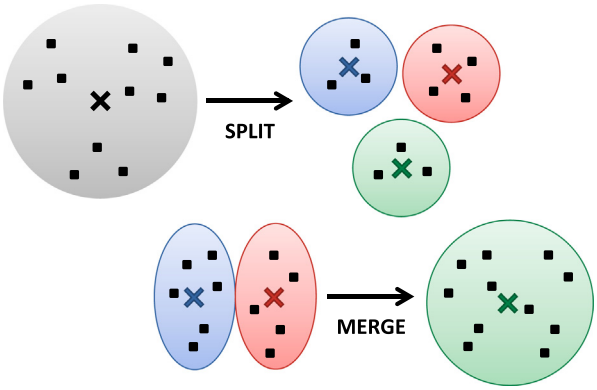

Fig. 6. Interactive refinement is carried out by repeated application of SPLIT and MERGE operations on the topic clusters.

if the automatic analysis has assigned multiple topics to the same cluster, which normally becomes obvious after a brief perusal of the most prototypical responses (as shown in Figs. 3 and 4).

The MERGE operation combines two or more topic clusters into a larger cluster with a single prototype. This step is appropriate if the automatic analysis has formed separate clusters with very similar meaning, sometimes because of differences in the wording of the responses. Suitable candidates for MERGE operations are often obvious from the cluster labels.

Each manual intervention is optionally followed by several iterations of the truncated k-means algorithm, so that documents from other clusters can also be reassigned based on the new information. In this way, the topic analysis can often be improved substantially within a few minutes. The effects of parameter optimization and interactive refinement are discussed and evaluated in Section 4.2.

3.4. Implementation

Our implementation of the TKE approach is based on tried and tested open-source technologies. The main programming languages are Python and R, each contributing its respective strengths. Python is used for text processing, linguistic annotation and sentiment analysis in the preprocessing module. It also serves as a "glue language" between the GUI and the analysis and refinement modules, taking care of input and output, format conversions and parameter settings.

The core parts of the analysis and refinement modules, which rely heavily on statistical algorithms, are implemented in the programming language R [21]. Beside its huge library of various statistical and data analysis functions, R offers efficient matrix algebra operations, including support for compact sparse matrix representations.

R was also essential in providing flexible, Web-based GUIs based on the Shiny framework [22]. These rapid-prototyping GUIs were used extensively in the development and testing of the TKE system. All screenshots in this paper show the development GUI implemented with R and Shiny.

Data exchange between TKE and the external GUI is performed through JSON files, a simple representation standard that can encode complex data structures, is human-readable and has relatively little overhead (compared e.g. to XML). Cache files used to exchange data between the modules are optimized for efficiency. The output of the preprocessing module is stored in a set of TAB-delimited text files, while the analysis and refinement modules store their complete internal state in binary, compressed RData (`.rda`) format.

4. Quantitative evaluation

This section is concerned with an evaluation of the TKE system in order to confirm the validity of its analysis results. We focus on sentiment analysis and topic clustering as the two core operations of TKE; these are also the most complex and innovative components of the system. Evaluation methods and results for each component are presented in the following two subsections.

It has to be said, though, that neither the evaluation of sentiment analysis nor that of topic clustering can be considered a straightforward task. The reason lies in the fact that TKE was developed to solve interpretative tasks that are deeply subjective and on which even human annotators often disagree. It is therefore not trivial to identify appropriate quantitative measures of TKE's analysis quality and to interpret the evaluation results.

4.1. Sentiment analysis

The German sentiment analysis module was trained and evaluated on in-domain data randomly selected from nine different opinion polls. Approx. 10,000 short responses to open questions were manually annotated by German native speakers, employed by Rogator, with respect to their positive, negative or neutral polarity. Annotators were instructed to focus on sentiment stated explicitly by the respondents and classify texts as neutral otherwise. Annotation was carried out on a scale ranging from −2 to +2 (cf. Table 1). For classifier training and evaluation, the judgements were reduced to three categories: positive (+2, +1), neutral (0) and negative (−1, −2).

A subset of 1800 responses was annotated independently by three judges, which are identified below by their initials *AK*, *LB* and *MB*. This subset allows us to determine the level of interrater agreement. As we will see, even human annotators differ in their polarity judgements quite frequently. It would be unrealistic to expect an automatic system to achieve close to 100% accuracy. For this reason, we will consider interrater agreement an upper limit against which the performance of TKE's sentiment analysis is compared.

Full agreement between all three judges was achieved only on 62.1% of the data (1118 of 1800 responses). Corrected for chance agreement, this results in a rather low Fleiss' kappa of $\kappa = 0.543$ (for details on measures of interrater agreement see [23]). We then computed pairwise agreement between the annotators using weighted Cohen's kappa as a chance-corrected measure, shown in Table 2. The higher level of observed agreement appears to be mainly due to the better chance that two annotators will randomly

Table 1
Polarity scale for manual annotation of training data.

Rating	Description
−2	Definitely negative
−1	Probably negative
0	No explicit sentiment expressed
+1	Probably positive
+2	Definitely positive

Table 2
Pairwise annotator agreement, showing weighted Cohen's kappa as a chance-corrected measure and the observed level of agreement. Both values are on a scale ranging from 0 to 1. The annotator pair with highest interrater agreement is indicated in bold font.

Annotator pairing	Cohen's κ	Observed
AK vs. LB	0.578	0.754
AK vs. MB	**0.638**	**0.758**
LB vs. MB	0.566	0.722

Table 3
Confusion matrix for annotator AK vs. annotator MB.

		MB		
		Negative	Neutral	Positive
AK	Negative	290	43	4
	Neutral	129	806	171
	Positive	6	82	269

select the same category: κ values are still unsatisfactorily low. Since these values suggest that annotator LB may be unreliable or may have interpreted the annotation guidelines in a different way, we concentrate on the comparison between AK and MB in what follows. According to Landis and Koch [24, p. 165], $\kappa > 0.6$ can be interpreted as "substantial agreement" between these two annotators.

The confusion matrix between AK and MB in Table 3 shows that the two annotators rarely make conflicting decisions (positive vs. negative). However, it seems difficult to distinguish between neutral and polar (positive or negative) responses. The matrix also reveals a difference in the bias towards neutral ratings: AK classified 1106 responses (61.4%) as neutral, whereas MB only classified 931 responses (51.7%) as neutral.

On the full training data of approx. 10,000 responses, the TKE polarity classifier achieved an accuracy of 69.6% measured by ten-fold cross-validation, which compares favourably with the observed agreement of 75.8% between annotators AK and MB.

The full TKE system uses a modified version of this classifier with an additional bias towards the neutral category: positive or negative polarity is only assigned if its posterior probability (i.e. the confidence of the classifier) is at least 60%; otherwise the response is considered neutral. This ensures that TKE does not give a misleading impression of strong sentiment towards a topic by guessing inaccurately, especially on out-of-domain data.

A final evaluation of the production version of the TKE polarity classifier was carried out on a separate test set of more than 19,000 responses that were manually annotated by a student assistant. Table 4 shows the evaluation results in terms of precision, recall and F_1-score separately for each category, as well as the frequency of the category in the test set (support). The bottom row shows weighted averages over all three categories.

We have clearly achieved our goal of ensuring high precision for responses labelled as positive ($P = 78.8\%$) or negative ($P = 90.1\%$), at the expense of lower recall. The rationale for this trade-off is that the sentiment analysis should label a response as positive or negative only if this can be done with high confidence, so as not to give a misleading impression of strong sentiment. The overall quality of the TKE sentiment module is still very good, with a weighted average F_1-score of 69.4%. To put these numbers into perspective, we also computed precision, recall and F_1-scores for a human annotator (AK), measured against another annotator (MB) as gold standard on the 1800 responses annotated by both coders.

Table 4
Evaluation results for the TKE sentiment analysis module on German data (left panel). For comparison, the right panel shows the performance that human annotator AK would have achieved if evaluated against annotator MB as a gold standard. Scores cannot be compared directly, since the evaluation was carried out on different test sets.

	TKE sentiment module				Human annotator (AK)			
	P	R	F_1	Support	P	R	F_1	Support
Positive	78.8	64.6	71.0	22.0	75.4	60.6	67.2	24.7
Neutral	46.8	82.1	59.6	26.4	72.9	86.6	79.1	51.7
Negative	90.1	62.4	73.7	51.6	86.1	68.2	76.1	23.6
Average$_w$	76.2	68.1	69.4		76.6	75.8	75.5	

The results show a remarkably similar pattern of high precision and relatively low recall for positive and negative responses. While one has to keep in mind that the two panels in Table 4 are based on different test sets,[4] the weighted average F_1-score of 75.5% indicates that TKE does not perform much worse than human sentiment annotation.

4.2. Topic clustering

As stated above, it is very difficult to carry out a precise quantitative evaluation of TKE's topic clustering module because of the subjective nature of the interpretation and manual coding of responses to an open question. According to experts from Rogator, human annotators often arrive at substantially different code plans for the same data set. Even when working to the same predetermined code plan, they disagree on the classification of individual responses as well as the overall frequency of each topic category. While there is no practicable alternative to a manually annotated gold standard as a basis for the quantitative evaluation, one has to keep in mind that perfect correspondence between the topic clustering and the gold standard cannot reasonably be expected. For the same reason, supervised training or parameter optimization guided by a gold standard data set may be counter-productive, tuning the system to reproduce the subjective decisions of one particular annotator.

For the experiments reported here, Rogator provided a German-language data set of 1166 responses to an open question from one of their online surveys, which was manually coded by an experienced annotator. The annotator developed a code plan of 24 categories grouped into 8 high-level topic groups. Each response was assigned to one or more of the 24 categories.[5]

We applied TKE to this data set to produce three topic clusterings representing different levels of manual intervention by the user:

1. AUTOMATIC: A completely automatic analysis with all parameters left at default setting, obtained in less than 5 s.
2. EXPERT: An improved topic clustering obtained after 10 min of experimentation with system parameters, including user-defined stopwords and synonym sets. This corresponds to the amount of time a commercial user would typically be willing to invest in fine-tuning the analysis. For the present evaluation, parameter optimization was performed by one of the authors, using trial & error based on interpretability of the semantic map plot and some inspection of the topic clusters. Users will be able to carry out a similar parameter optimization without technical knowledge of the TKE implementation or the underlying algorithms.[6]
3. REFINED: An interactively refined topic clustering obtained by the application of SPLIT and MERGE operations for another 10 min, based on the EXPERT clustering. This represents the optimal analysis that a moderately experienced user can produce in 20 min or less of work, a reasonable amount of time for an analyst to spend on this task. Note that the time required does

[4] In particular, the test set used for the final evaluation of the TKE sentiment module contains a much higher proportion of negative responses, mainly because it was compiled from responses to different open questions.

[5] 64 responses were not assigned to any categories, presumably an oversight of the human annotator.

[6] The following parameters were changed in this step: (i) The number of clusters was reduced from 20 to 15. (ii) The number of latent SVD dimensions was increased from 35 to 40. (iii) 11 user-defined stopwords were added (e.g. *sehr* 'much', *finde* 'I think' and *leider* 'unfortunately'). (iv) Two sets of synonyms were defined: *Video*, *Videos* (because the stemmer failed to recognize the plural form) and *Markenauswahl*, *Auswahl (von) Marken* (different morpho-syntactic realizations of the same concept 'choice of brands'; automatic compound splitting is a notoriously difficult problem in German).

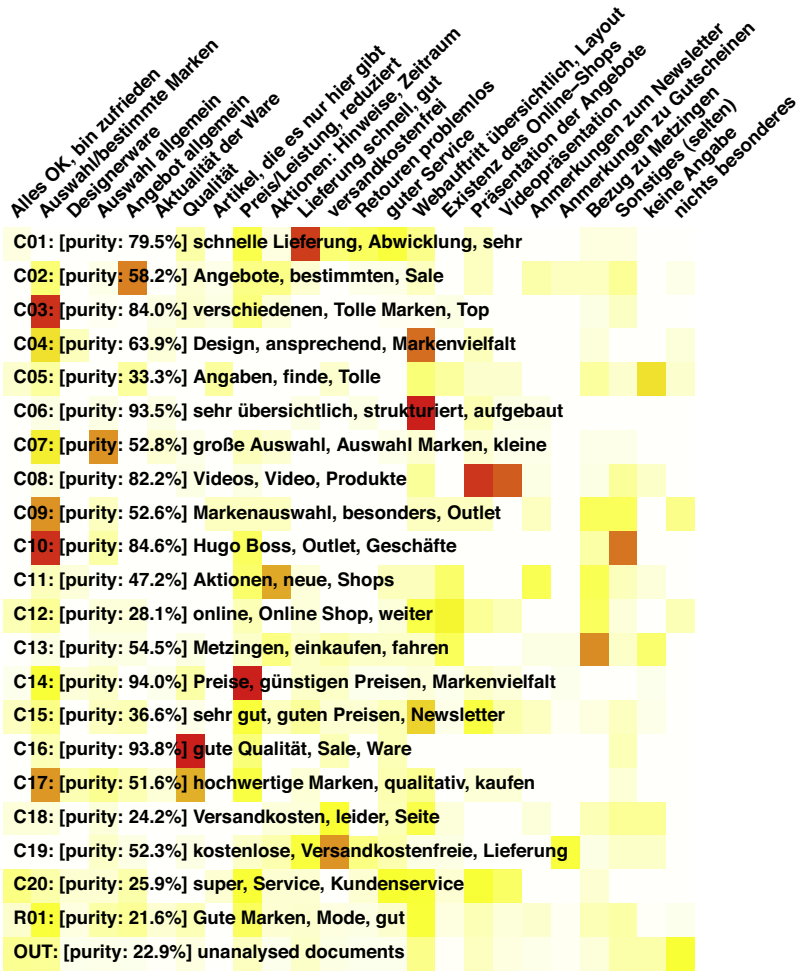

Fig. 7. Agreement of TKE topic clusters (rows) with gold standard category assignments (columns), for a fully AUTOMATIC clustering.

not depend on the number of responses: the larger a data set, the more favourable the comparison with a manual coding procedure becomes.

All experiments were carried out with the Web-based GUI used for development of the TKE system.

A first indication of the quality of TKE's topic clustering is the comparison of topic clusters (rows) with gold standard categories (columns) visualized in Fig. 7. Each cell of the plot shows how many of the responses in a given cluster were assigned to a given topic category by the human coder. A dark red colour indicates a proportion close to 100%, i.e. the cluster corresponds perfectly to this topic category (or a subset of the category).[7] White cells indicate topic categories that do not occur at all in a cluster, and orange corresponds to a proportion around 50%.

Several interesting observations can immediately be seen in Fig. 7. About a third of the topic clusters (C01, C03, C04, C06, C08, C09, C10, C14, C16) correspond very well to a gold standard category and could easily be labelled accordingly by users of the system. As the gold standard allows multiple category assignment, one cannot expect a one-to-one correspondence between clusters and topic categories. For example, C08 reflects the fact that respondents commenting on the presentation of goods in a Web shop (*Präsentation der Angebote*) usually also mention the product videos (*Videopräsentation*). It is therefore appropriate to group these responses into a single topic cluster. In fact, the usual numeric summary of a manual coding, showing only the number of responses for each topic category, would give an incomplete picture because it fails to show the correlation between the two categories.

Some categories are split across multiple topic clusters, e.g. range of brands (*Auswahl/bestimmte Marken*) across clusters C03, C09, C10 and C17. Again, this is an appropriate subdivision focusing on different aspects of the topic category, which is expressed by multiple assignments in the gold standard. Responses in C17 praise both the range and quality (*Qualität*) of brands, and responses in C10 focus on a particular German brand, which is hidden in a category labelled "infrequent other topics" (*Sonstiges (selten)*) by the manual coding. C03 and C09, on the other hand, represent different linguistic expressions of the same meaning (*Markenauswahl* vs. *verschiedene, tolle Marken*, both referring to the range of brands). Because TKE does not rely on expensive ontological or lexical resources, it cannot recognize that both clusters belong to the same topic category. However, this is obvious to any native speaker of German from the cluster labels and a cursory inspection of the corresponding responses, so it can easily be taken into account in users' interpretation of the TKE results.

Responses that could not be assigned clearly to one of the topic clusters are collected in a residual cluster R01 here. As expected, they are evenly distributed across most of the topic categories in the gold standard. The same holds for unanalyzed responses, shown in the bottom row (OUT) of the display.

[7] Note that the darkest cell in a row corresponds to the purity value shown in the plot. For example, 93.5% of the responses in cluster C06 were assigned to the topic labelled *Webauftritt übersichtlich, Layout*.

The agreement between the TKE analysis and the gold standard is quantified by a variant of the widely-used purity measure [25, p. 357]. Each cluster is automatically labelled with the most frequent gold standard category present in the cluster, which represents an interpretation of the clustering that is as close to the manual coding as possible. The purity of a cluster is the proportion of its responses that are labelled correctly, i.e. that have been assigned to this category in the gold standard. Purity values for the individual clusters are shown in square brackets in Fig. 7. The overall purity is the weighted average of purity values across all regular clusters (excluding R01 and OUT).

Purity has some drawbacks as an evaluation criterion: it cannot account properly for multiple assignments (as is the case for our data), and its value depends on the number of clusters generated

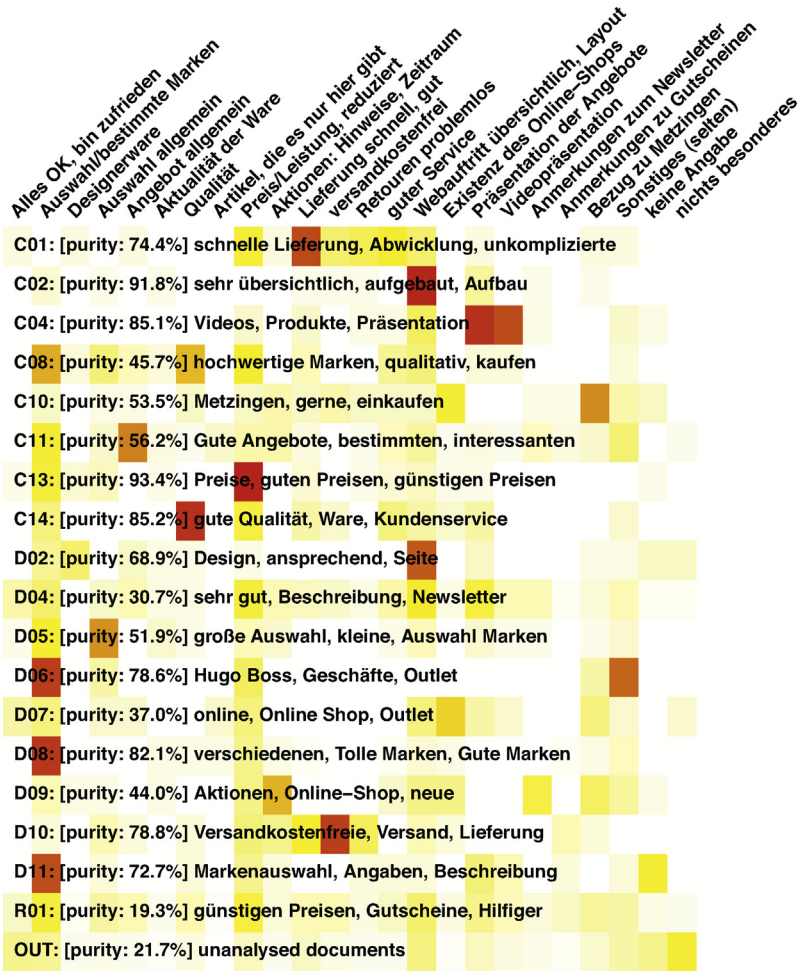

Fig. 8. Agreement of TKE topic clusters (rows) with gold standard category assignments (columns) after parameter optimization and interactive refinement (REFINED clustering).

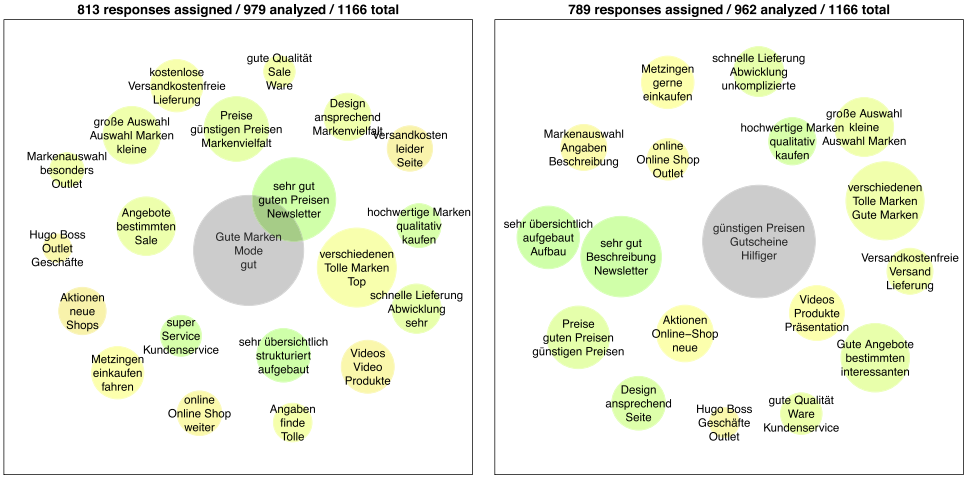

Fig. 9. Semantic map of topic clusters (left panel: fully AUTOMATIC analysis, right panel: interactively REFINED solution).

Table 5
Weighted average purity (%) of regular topic clusters for the fully AUTOMATIC clustering, optimized parameters (EXPERT) and interactive refinement (REFINED). Each clustering is evaluated against the full 24 categories of the manual code plan (*fine*) as well as 8 high-level topic groups obtained by combining similar categories (*coarse*). For comparison, the purity of the residual cluster R01 is also shown.

	# clusters	Fine (24 categories)		Coarse (8 groups)	
		Regular	R01	Regular	R01
AUTOMATIC	20	60.78	21.65	71.33	42.27
EXPERT	15	61.85	23.68	74.20	42.11
REFINED	17	65.16	19.29	75.18	38.07

by the system.[8] However, unlike many other evaluation criteria, it can be interpreted intuitively as the overall accuracy of the automatic topic analysis, i.e. the proportion of responses assigned to the correct topic category (if clusters are labelled appropriately).

Table 5 shows that the fully automatic TKE analysis achieves a purity of 60.78% across 20 clusters.[9] With 10 min of parameter optimization, purity can be increased to 61.85%; with another 10 min of interactive refinement, if is further increased to 65.16%. At the same time, the number of clusters is reduced (as shown in the second column), leading to a simpler and more accurate analysis of the data set. Fig. 8 visualizes the better correspondence between TKE clusters and manually assigned topic categories after interactive refinement.

Fig. 9 illustrates that the higher purity and lower number of clusters of the REFINED analysis indeed translates into a clearer and more interpretable topic map (right panel, compared to the fully AUTOMATIC analysis in the left panel).

In market research, the analysis of an open question is often summarized in the form of an overview of identified topic groups and their frequency, i.e. the proportion of responses that address each topic. Fig. 10 compares a topic overview for the manual coding in the gold standard (grey bars) to an overview based on the REFINED TKE analysis with optimal cluster labels (blue bars). Though far from perfect, the TKE analysis gives a reasonably good impression of the most important topics in the data set and their relative frequency. The topic categories that TKE failed to identify are infrequent, occurring in less than 5% of responses each. Note that the first three categories at the top of the plot are really non-topics, labelled as "nothing special" (*nichts besonderes*), "cannot say" (*keine Angabe*), and "other" (*Sonstiges*). It is therefore appropriate that TKE assigned them to the residual cluster R01, which is not included in the topic overview.

At the level of general topic groups, which are less prone to subjective differences, purity is considerably higher and ranges from 71.33% to 75.18% (right-hand columns of Table 5). The topic overview in Fig. 11 shows that the topic frequencies computed by TKE match the gold standard frequencies very well at this coarse level. In this case, responses assigned to the residual cluster R01 (197 out of 1166 responses = 16.9%) have been included in the bar for the residual category (*Sonstiges, keine Angaben*).

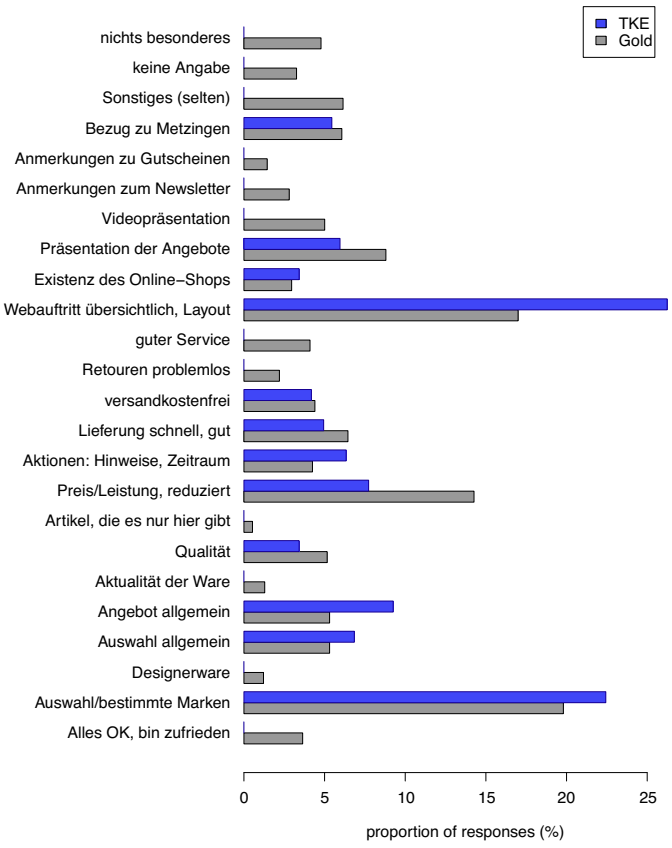

Fig. 10. Overview of topic categories and their frequency for REFINED clustering (blue bars), compared to gold standard (grey bars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

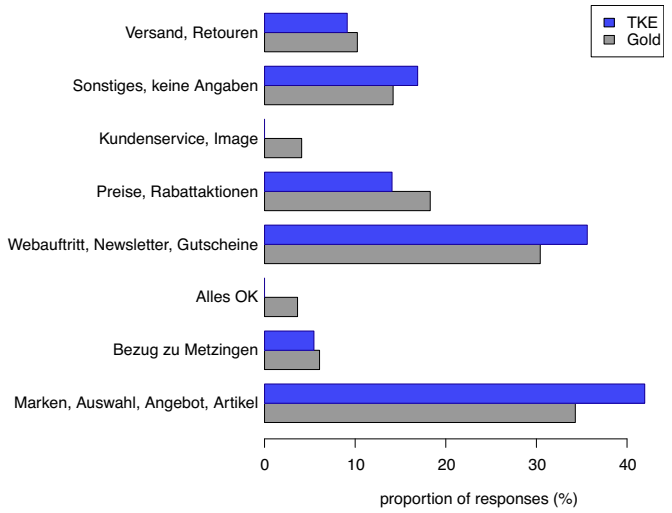

Fig. 11. Comparison of TKE vs. gold standard topic overviews at the coarse level of 8 general topic groups.

5. Conclusion

TKE is a versatile tool for the identification of topics in medium to large sets of short text documents, as well as the general sentiment expressed towards each topic. From a user point of view, the main strength lies in its quick and therefore cost-efficient way of analysis. The system copes easily with amounts of material that render manual processing virtually impossible. With the increasing popularity of online surveys (and other applications such as

[8] A more fine-grained clustering usually leads to higher purity, because the chance that all responses in a cluster belong to the same gold standard category (if only by accident) increases. As an extreme example, assigning each response to its own unique cluster would result in a purity value of 100%. Therefore, the number of clusters has to be taken into account if clusterings of different granularity are to be compared. Note that the EXPERT and REFINED clusterings achieve higher purity with a smaller number of clusters than the AUTOMATIC analysis.

[9] Note that the number of clusters is not equal to the number of categories in the gold standard. Our evaluation tests a realistic setting in which the automatic analysis and manual refinement are carried out without any knowledge of a manual code plan for the data set.

trend mining in the Web or social media networks) data sets of this size are no longer uncommon.

Applied in a purely automatic fashion, TKE is able to deliver a quick, but comprehensive overview of the main topics and sentiments in a text collection. With semi-automatic optimization by parameter adjustments and interactive refinement, these initial results can be fine-tuned until a clearly interpretable and accurate solution is achieved. As shown in Section 4.2, an analysis carried out in this way delivers results similar to those produced by human coders, but in a considerably shorter time span and at a much lower cost. Furthermore, TKE provides concise graphical and tabular summaries of its results without any additional expenditure.

By relying heavily on a statistical analysis of the information contained within the input data instead of expensive knowledge resources, TKE is largely domain- and language-independent. Adaptation to and analysis of previously unseen questions is possible out-of-the-box in most cases, omitting the less crucial sentiment analysis component if necessary.

TKE thus provides a potent instrument for the analysis of textual data in market research and shows promising potential for a range of similar applications.

Acknowledgements

We would like to thank two anonymous reviewers for their valuable comments, which helped to improve the final version of the paper with a better description of the system and the evaluation experiments.

References

[1] W. Zhang, T. Yoshida, X. Tang, Q. Wang, Text clustering using frequent itemsets, Knowl. Based Syst. 23 (2010) 379–388.
[2] D.M. Blei, A.Y. Ng, I. Jordan, Michael, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[3] Y. Li, S.M. Chung, Text document clustering based on frequent word sequences, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management CIKM '05, ACM, New York, NY, USA, 2005, pp. 293–294.
[4] F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '02, ACM, New York, NY, USA, 2002, pp. 436–442.
[5] B. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: Proceedings of the SIAM International Conference on Data Mining 2003 (SDM 2003), 2003.
[6] P. Cimiano, J. Völker, Ontology construction, in: N. Indurkhya, F.J. Damerau (Eds.), Natural Language Processing, Chapman and Hall, London, New York, 2010, pp. 577–604.
[7] G. Bordea, S. Kirrane, P. Buitelaar, B. Pereira, Expertise mining for enterprise content management, in: N.C.C. Chair, K. Choukri, T. Declerck, M.U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12), European Language Resources Association (ELRA), Istanbul, Turkey, 2012.
[8] B. Fortuna, M. Grobelnik, D. Mladenic, OntoGen: semi-automatic ontology editor, in: M.J. Smith, G. Salvendy (Eds.), Human Interface and the Management of Information. Interacting in Information Environments. Volume 4558 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2007, pp. 309–318.
[9] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.
[10] M.F. Porter, An algorithm for suffix stripping, Program 14 (1980) 130–137.
[11] T. Proisl, P. Greiner, S. Evert, B. Kabashi, KLUE: Simple and robust methods for polarity classification, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), vol. 2. Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, GA, (2013), pp. 395–401.
[12] S. Evert, T. Proisl, P. Greiner, B. Kabashi, SentiKLUE: updating a polarity classifier in 48 hours, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014), Dublin, Ireland, 2014.
[13] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (1975) 613–620.
[14] P.D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics, J. Artif. Intell. Res. 37 (2010) 141–188.
[15] J.R. Bellegarda, Latent semantic mapping: principles & applications, Synth. Lect. Speech Audio Process. 3 (2007) 1–101.
[16] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, New York, 1990.
[17] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[18] P. Rayson, R. Garside, Comparing corpora using frequency profiling, in: Proceedings of the ACL Workshop on Comparing Corpora, Hong Kong, (2000), pp. 1–6.
[19] T.E. Dunning, Accurate methods for the statistics of surprise and coincidence, Comput. Linguist. 19 (1993) 61–74.
[20] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S-PLUS, 4th ed., Springer, New York, 2002.
[21] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2010.
[22] RStudio, Inc., shiny: Web Application Framework for R, 2014 URL: http://CRAN.R-project.org/package=shiny, R Package, version 0.10.
[23] R. Artstein, M. Poesio, Survey article: inter-coder agreement for computational linguistics, Comput. Linguist. 34 (2008) 555–596.
[24] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174.
[25] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

Stefan Evert is professor of Corpus Linguistics at Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. He earned a PhD in Computational Linguistics from the University of Stuttgart in 2004 and held a position as assistant professor for Computational Linguistics at the Institute of Cognitive Science, University of Osnabrück from 2005 to 2011. From 2011 to 2012 he was professor of English Computational Corpus Linguistics at Technische Universität Darmstadt, Germany. His main interests lie at the boundary between linguistic research, statistical corpus analysis and natural language processing. Current research topics include the methodological foundations of corpus linguistics, collocations and multiword expressions, distributional semantics and multi-dimensional analysis of language variation.

Paul Greiner is a PhD student and research associate with teaching assignment at the professorship for Corpus Linguistics of the Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. He graduated at the same institute in Computational Linguistics and English Linguistics in 2012. His professional interest lies in everything concerning natural language processing in research as well as real world application, specifically identification and measurement of semantic textual similarity and polarity classification. During his studies, he focused on rule-based systems, performing the shift to statistical techniques in the recent years. His non-professional interests are manifold, ranging from vertical sports to playing four- to six-stringed musical instruments, among other things.

João Filipe Baigger is senior consultant at Rogator, a company specialized in market research software & consulting in Nuremberg, Germany. He studied pedagogy and psychology (diploma in 2001) and business administration (diploma in 2008). Since 2006 he works for Rogator in the market research department.

Bastian Lang is head of software development at Rogator, a company specialized in market research software & consulting in Nuremberg, Germany. He studied physics and graduated with a diploma in 2010 at Friedrich-Alexander-Universität Erlangen-Nürnberg in Germany. Since 2011 he is responsible for all software products of Rogator and their further or new development.