A Corpus-Based Method for Product Feature Ranking for Interactive Question Answering Systems

Natalia Konstantinova, Constantin Orăsan, and Pedro Paulo Balage

Research Group in Computational Linguistics
University of Wolverhampton
Wolverhampton, UK
N.Konstantinova@wlv.ac.uk and C.Orasan@wlv.ac.uk and pedrobalage@gmail.com

Abstract. At times choosing a product can be a difficult task due to the fact that customers need to consider many features before they can reach a decision. Interactive question answering (IQA) systems can help customers in this process, by answering questions about products and initiating a dialogue with the customer when their needs are not clearly defined. For this purpose we propose a corpus-based method for weighting the importance of product features depending on how likely they are to be of interest for a user. By using this method, we hope that users can select the desired product in an optimal way. For the experiments a corpus of user reviews is used, the assumption being that the features mentioned in a review are probably more important for a person who is likely to purchase a product. In an attempt to improve the method, a sentiment classification system is also employed in order to distinguish between features mentioned in positive and negative contexts. Evaluation shows that the ranking method which incorporates this information is one of the best performing ones.

1 Introduction

Every day millions of people are confronted with a task of choosing a new product. Given how fast the technology is progressing, it is becoming more and more difficult to make this choice without any additional help. Some prefer using face-to-face communication with shop assistants, but what if this option is not available or you prefer online shopping? Interactive question answering (IQA) systems can assist customers in this process by providing answers to the questions about products and initiating a dialogue with the customers when their needs are not clearly defined. This is particularly relevant when a product has many features that need to be considered before buying it.

The use of an IQA system should facilitate the process of choosing a product and make it more efficient. However, in order to achieve this goal, the system should know which features are more important for the users and as a result should be given a priority in the decision process. Therefore it is essential to find some ranking methods to assist the navigation of the available features.

This paper proposes a corpus-based method for weighting the importance of product features using a corpus of reviews. Our assumption is that these texts will contain references to product features and, because they are reviews, they will focus on the features that are more likely to determine the purchase of the product. We test our method in the domain of mobile phones, however we believe that it can be employed to other domains describing products.

The remainder of the paper is structured as follows: The next section (Section 2) discusses the related work in the field. Section 3 presents the description of the experiment including its justification and ranking methods developed. The evaluation of the system, including the gold standard, evaluation metrics used and error analysis is presented in Section 4. The paper finishes with conclusions and directions for future work (Section 5).

2 Related Work

This paper addresses the problem of content management for interactive question answering systems which is related to dialogue managers which are part of dialogue systems. However, to the best of our knowledge there is no research similar to the one carried out in this paper. In addition, the novelty of our approach also comes from the fact that it lies at the intersection of several NLP fields such as information extraction, IQA and sentiment analysis. However, work in information extraction, sentiment analysis and interactive question answering can be considered as the most relevant to our research and is briefly presented next.

There are a number of projects focusing on feature extraction for sentiment analysis. The system described in [3] extracts opinion summary about products, but instead of getting the opinion about the product in general, the proposed method tries to produce an opinion summary about separate features. For this purpose, they mine product features discussed by the customers and rate each opinion as positive or negative. Later this information is used to produce feature-based summaries about the products. Authors of [7] aim at solving a similar problem but use multiple specifications of a product for further clustering and extracting of product features. It is also done in order to produce summaries describing the products. Opine [8] is an example of an unsupervised information-extraction system which mines reviews in order to build a model of important product features, their evaluation by reviewers, and their relative quality across products.

Different approaches were developed to address the problem of feature extraction - unsupervised [10] and semi-supervised methods [15], as well as topic modeling [16]. Some researches tried to build specialized domain ontologies manually in order to get better quality resources, but we are aware of only one ontology describing mobile phones [4].

Our research differs from the aforementioned works, because we do not focus on extracting features of the phone from the reviews. Instead, we are more interested in ranking already acquired lists of features using the available

customer reviews. In this respect and keeping in mind the goal of our research, it is worth mentioning work previously done in the field of IQA.

Even though we are also not aware of applications in the field of IQA similar to our research, there are several IQA systems that address the problem of effective information management which can be considered relevant to work in question. These systems attempt to help users choose products and rely on constraint-based approaches ([9], [14], [11]).

The system developed by Yan Qu and Nancy Green [9] focuses on providing airline flight information. After the user fills in all the constraints, the system submits a query to the database and if the request was over-constrained, ways to relax the constraints are suggested. In contrast to our method, they do not employ any initial ranking of the constraints in order to set priorities for the use of some constraints.

The approach developed by [14] is used in several systems dealing with restaurant selection, MP3 player operation and navigation tasks. Their aim is to find efficient ways of managing a dialogue and providing a sufficient amount of information to users so that they are neither overwhelmed with too much information, nor left uncertain about some details. Their goal is to choose a single item out of a larger set of items, which is similar to the task we are dealing with. They focus on content optimization where dialogue strategies for dealing with query results are used. Rules governing dialogue steps based on the amount of results are manually constructed and thresholds are predefined. Even though the task is similar to choosing a product (a task we are addressing in this paper), they do not give priority to any of the features and use only quantitative information to make a decision about constraint relaxation and further suggestions. However, in this approach an ontology describing the data and the constraints is used, but constraints do not have any internal ranking.

The research in [11] is similar to [14], however they use predefined rules and simulate interactions with the user in order to further use this information for learning the best policies. They study the domain of in-car and in-home applications and provide examples of dialogues for choosing a song for a playlist. They handle the situations of under-constrained or over-constrained requests and learn ways to deal with them. However, the paper does not mention anything about the usage of an initial ranking of the constraints which we believe can help in the task of IQA.

The systems described above focus on the interaction in terms of the constraint-based systems, however none of them tries to rank the constraints or propose methods to make search for information more optimal in this way. In all the cases, either hand crafted or learnt policies are used to decide which dialogue move to take next. These systems try to act according the number of results they get, and on the basis of this information they attempt to relax the request or ask for additional constraints. We are more interested in suggesting new constraints to the customer and would like to select those that will help the user to choose a product in the quickest time. However, this aspect of the problem is not discussed in these research works.

3 Experiment

Given that our aim is to optimise the process of selecting a product on the basis of its features using an IQA approach, we evaluated several methods for ranking features. These methods are presented in this section and evaluated in Section 4. We start this section by providing a justification for the experiment carried out here.

3.1 Justification of the experiment

One way to identify the importance a feature plays in choosing a product is to collect and analyse a large number of interactions between a human and a sales assistant or a computer. This information can be used to learn the appropriate ranking of the features. However, this approach is labour intensive and time consuming, which makes it very expensive especially because it is domain dependent. As a result of the domain dependency, information gathering needs to be repeated every time a system is adapted for a new domain. For this reason, we propose a method which relies on user reviews to determine this ranking.

The underlying assumption of this method is that the most important features will also be mentioned frequently in the user reviews. Therefore, we believe it is possible to propose several weighting schemes which take a corpus of reviews and produce the ranking. Given that these reviews contain a large number of opinionated sentences, NLP techniques are being used to differentiate between positive and negative sentences. This is done in order to identify whether certain types of sentence (e.g. positive) are more likely to contain the necessary information to rank the features correctly.

3.2 Ranking methods

We developed several methods for the ranking of product features on the basis of their occurrence in our corpus. In this paper we use features of mobile phones, but the method can be adapted to any other products.

To carry out the experiments presented in this paper, we had to first identify features that can be of interest for users and therefore need to be ranked. Manual construction of such a list did not seem objective enough, and therefore we relied on the infoboxes present in Wikipedia pages describing products of the type of interest (i.e. in the case of this research pages describing mobile phones). The infoboxes contain brief tabular information summarising the content of the page and in the case of products quite often refer to the features of a product. For example, infoboxes in pages about phones may contain the feature "camera" and its corresponding value "5 megapixels". By collecting these features, we built our list to be ranked. The values corresponding to the features were also collected as a way of identifying indirect references to the features in the text.

Ways to match features: Once we managed to collect the list of features, we could investigate ways to rank them. It was decided to use NLP techniques to find the best ranking algorithm in order to avoid spending a significant amount of time and resources on collecting real customer interactions.

Given the fact that a feature can be expressed in several ways, we used several methods for matching the features extracted from the Wikipedia infoboxes with their occurrences in the texts. For all the ranking methods described in the next subsection, three types of matching methods were used:

- surface-based (also referred to as strict match),
- fuzzy matching (e.g. *battery life* and *lifespan*),
- values for features (e.g. *5 megapixels* and *camera*).

Surface matching implies a strict match between the string denoting a feature from the Wikipedia infoboxes and a string in the corpus. This matching technique does not allow any flexibility on how the feature is expressed in the text. Therefore this type of matching brings some limitations, as language is ambiguous and there are many ways to express the same thing using different surface representations. For this reason, we also implemented a *fuzzy matching* method which takes into consideration not only the surface form, but also considers synonyms extracted from WordNet [2] and manually compiled lists. Several of the problems identified with the first method were solved using fuzzy matching and are discussed in Section 4.6. At the same time, fuzzy matching introduces its own errors which are discussed in the same section.

Another way to improve the matching algorithms is to consider that a feature occurs in a text not only when it is directly mentioned, but also when values corresponding to a feature are used. Despite the appeal of this approach, there are values which are multiword expressions, so a strict matching would give a very low recall. For this reason, we used heuristics which consider a match successful if at least 60% of the text denoting a value was found. This helped us identify more information, but revealed the problem of overlapping features which will be further discussed in Section 4.6.

Frequency-based ranking: The first method explored relies on the frequency of a feature in our corpus of reviews in order to determine its importance. The assumption here is that the more frequently a feature is mentioned, the more important it is for the users. This approach was inspired by automatic summarisation [5]. Therefore, we extract frequency of each particular feature mentioned and use it as its score. For this purpose, all three different types of matching mentioned in the previous subsection were used.

Opinion-based ranking: Given that we are dealing with a corpus of reviews, we thought it could be beneficial to use the polarity of the sentences contained in the reviews in the ranking process. For determining the polarity of a sentence, we use a lexicon-based algorithm based on the SO-CAL algorithm [12]. This method relies on a dictionary which contains words and their semantic orientation scores

related to the sentiment expressed. This semantic orientation ranges from -4 to 4, where -4 stands for a totally negative word and 4 for a totally positive word. For our experiments we use the dictionary developed for the original method [13].

In the above mentioned method, the polarity of a sentence is measured as the sum of the semantic orientation present in the words. Those words and their part-of-speech are checked in the dictionary and the semantic orientation computed. Negation markers, modals and intensifiers change the polarity for the next word. The sentence is labeled as positive or negative if the overall semantic orientation is positive or negative. The sentences with score 0 are labeled as being neutral.

We developed two ranking methods based on the identification of the opinion in the text. The first of them takes into account only opinionated sentences and ignores the neutral ones. The assumption here is that the authors of the reviews will express opinions (positive or negative) about features which they find important to them. Frequency-based ranking was applied to the sentences that contain sentiment information. However, we should mention that we did not attach opinions to the particular features and just identified them at the sentence level.

Given that neutral sentences may contain information that could be useful for the ranking, a weighted ranking method which relies on opinion information was implemented. In this method, each occurrence of a feature in a neutral sentence receives a score of only 0.5, whereas an occurrence in an opinionated sentence gets a score of 1. In addition, two more experiments were run which considered only the positive and negative sentences for computing the ranking.

The next section presents the results of the evaluation.

4 Evaluation

4.1 Corpus description

For the experiments reported in this paper we compiled a corpus of reviews from the Epinions.com[1] website. Our research focuses on the development of an IQA system for mobile phones. For this reason, we collected reviews from the category *Cellular Phones* on the 21st October 2011. Our corpus contains 3,392 reviews (114,708 sentences) organised into two labels: "yes" and "no". These labels reflect the user's opinion about the product; whether the product is recommended or not. We have 2,437 reviews with the label "yes" and 955 with the label "no", but at this stage we do not use the user's opinion for the ranking. The method used to collect the corpus ensures that the approach can be easily applied to other domains.

[1] http://www.epinions.com/

90	price
81	battery
57	operating system
52	phone style
42	manufacturer
31	size
29	standby time
29	GPS
25	connectivity
24	3g network speed
23	memory
22	network data connectivity
21	camera
21	talk time
20	weight
19	keyboard
19	main screen
19	touchpad
18	CPU
18	hardware platform

Table 1. The top 20 features together with their frequencies

4.2 Gold standard

For the evaluation purposes we conducted a separate experiment which was aimed at constructing the gold standard. We wanted to use humans' input to rank the features they find most important when choosing a phone. For this purpose, we developed a special drag-and-drop interface which allowed the users to choose the most important features. No special guidelines were given to participants except that they need to pick the 5 most important features for them from a given list. The features were displayed in random order.

In order to prepare the initial list for ranking, we manually checked the features collected from Wikipedia infoboxes and removed those that were difficult to understand without further explanation. We also had to limit the number of options we showed to a user, so that the interface stayed user-friendly and easy to use. For this reason, after discussion between the authors of this paper, it was decided to keep only 47 features we felt to be the most important. We collected a total of 170 answers and used this information to get a weighted ranked list of features by assigning to each feature a score that is equal to the number of times a feature was selected. Table 1 shows the top 20 features together with their frequencies.

4.3 Baseline

In order to evaluate how effective our ranking methods are, we implemented two baselines. The first baseline considers the information from the infoboxes

and ranks a feature on the basis of how many Wikipedia articles about mobile phones mention the feature and assigns it a value. The second baseline ranks a feature on the basis of how many times it is mentioned in the Wikipedia articles describing mobile phones. By using these baselines, we can see whether a corpus of reviews is beneficial to us.

4.4 Evaluation metrics

Our evaluation was based on comparing several rankings to each other, so we had to consider some formal metrics which will give us an objective number. We decided to choose two metrics that are commonly used to measure the association between two measured quantities.

The first one is the Kendall rank correlation coefficient and is commonly referred to as Kendall's tau coefficient [1]. It depends on the number of inversions of pairs of objects which would be needed to transform one rank order into the other [1]. Equation 1 describes the formula used for calculating Kendall rank correlation coefficient.

$$\tau = \frac{N_c - N_d}{\frac{1}{2} * n * (n-1)} \tag{1}$$

where N_c is the number of concordant pairs, N_d is the number of discordant pairs, whilst n is the total number of pairs. τ takes values between -1 and 1, where -1 means that two rankings are the reverse of each other and 1 shows that rankings are the same.

The second metric we used is Spearman's rank correlation coefficient or Spearman's rho and is a non-parametric measure of statistical dependence between two variables [6]. Spearman's rank takes into account differences between the ranks of each observation on the two variables and Equation 2 shows the way this metric can be calculated.

$$\rho = \frac{\sum_i (x_i - \overline{x}) * (y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 * (y_i - \overline{y})^2}} \tag{2}$$

Similar to the Kendall's tau, the Spearman's rho values range from -1 to +1, and the closer to +1 they are, the more similar the rankings are. The use of these metrics allowed us to output a score after comparing two lists and the results will be provided in the next section.

4.5 Results

As described in the previous sections, we have carried out several experiments to produce different rankings of the features. We compared our rankings to the gold standard and the results of this comparison can be found in Table 2. For the evaluation, we used both the full gold standard and only the first 20 items in the gold standard. The justification for the second list is that it is highly unlikely

Method	Full list		Top 20 items	
	τ	ρ	τ	ρ
Baseline_1	-0.084	-0.155	0.017	0.019
Baseline_2	0.009	0.010	0.146	0.177
Frequency from reviews exact match	0.220	0.326	0.187	0.252
Frequency from reviews fuzzy match	0.116	0.164	0.209	0.292
Frequency from reviews values match	0.245	0.357	0.470	0.612
Frequency from opinionated sentences exact match	0.218	0.326	0.241	0.357
Frequency from opinionated sentences fuzzy match	0.165	0.245	0.209	0.317
Frequency from opinionated sentences values match	0.241	0.346	0.513	0.647
Weighted frequency with exact match	0.159	0.235	0.166	0.211
Weighted frequency with fuzzy match	0.207	0.298	0.230	0.332
Weighted frequency with values match	0.051	0.083	0.123	0.160
Frequency from negative sentences with exact match	-0.051	-0.078	0.016	0.008
Frequency from negative sentences with fuzzy match	-0.115	-0.172	-0.026	-0.056
Frequency from negative sentences with values match	-0.024	-0.045	-0.053	0.134
Frequency from positive sentences with exact match	0.011	0.018	0.123	0.130
Frequency from positive sentences with fuzzy match	0.175	0.253	0.155	0.222
Frequency from positive sentences with value match	0.125	0.197	0.021	0.000

Table 2. The evaluation results

that a customer will be willing to be asked about more than 20 features before they reach a decision.[2]

In Table 2 the rows $Baseline_1$ and $Baseline_2$ correspond to the two baselines introduced in Section 4.3. As can be seen, the results obtained with the two baselines are among the lowest indicating that using a corpus such as Wikipedia articles is not useful. The three rows with labels starting with *Frequency from reviews* contain the results obtained using just frequency of features in the reviews, but employing different feature matching method. The remainder of the rows contain the results of the methods that use the opinion classifier and different feature matching methods.

4.6 Error analysis

Analysis of the results reveals that the best performing methods are the ones using either the frequency of the features in the full corpus of reviews or the frequency of features only in the opinionated sentences. In both cases, the values of the features are used for matching. These results hold both for the full gold standard and when only the top 20 items are considered. The rest of the results are considerably lower, especially when only the top 20 items are considered. Strangely enough, the method which gives a weight of 0.5 to features that appear in sentences that do not have a polarity, and which is somehow between the two

[2] In reality, we hope that by using the ranking methods presented in this paper and the interactive question answering system that we are currently developing, the number of questions will be much lower.

best performing methods in terms of how features are scored, performs rather poorly regardless of the matching method used. The same happens if we use only the positive or negative sentences.

The experiments carried out in this paper revealed several problems to be addressed in order to get better results. One of the first issues we had to address when implementing the matching algorithm was the possibility to refer to the same feature in several different ways. For example the feature *operating system* can be referred to using "Operating System", "operatingsystem" or "os". Even though we used WordNet and manually compiled lists, it is unlikely that we managed to cover all the possible ways people use to refer to a feature. For this reason, the fuzzy matching method is not always very precise. Related to this problem is the fact that the list of values of a feature is likely to grow over time. Unless these values are listed in Wikipedia and our matching algorithm gets updated there is no way to capture the mention of a corresponding feature in a review.

Another problem with the approach described in this paper is related to the ambiguity of the features. For example, the features "standby time" and "usage time" have very similar meaning. This situation becomes even more problematic when the features are considered out of the context, as in the case of the experiment carried out to produce the gold standard. In light of this, word sense disambiguation-like methods could be considered to find out whether two similar expressions refer to the same feature on the basis of their context.

Another problem related to matching of features is with pairs such as "camera" and "video camera". When using only strict matching, it is difficult to decide whether the users just described a photo camera or whether they are referring to a photo-video camera. This problem becomes more serious when both forms are used in the text and "camera" is coreferential with "video camera". The only way to address this problem is to employ a coreference resolver.

The use of WordNet to obtain synonyms introduced a fair amount of errors as well. For example, for the feature "carrier" some of the synonyms are "postman", "carrier wave", "mailman" and "attack aircraft carrier" which are completely unrelated to the features of mobile phones. This is due to the fact that the word used to refer to this feature is far too general and therefore ambiguous. At the other extreme are the features such as "hardware platform" which are too specific and do not appear in WordNet. For this reason, it will be necessary to produce a better list of synonyms for the features.

5 Conclusions and Future Work

This paper addressed the problem of feature ranking for interactive question answering systems which help customers to choose the right product for them. Two baselines and several ranking methods were evaluated against a gold standard collected from users. The Kendall rank correlation and the Spearman's rank correlation coefficients were applied in order to provide an objective evaluation of the ranking methods applied. An experiment showed that two of

the ranking methods proposed perform far better than any other methods. The evaluation also confirmed the fact that using a corpus of reviews is beneficial for feature ranking. The results were further improved by using only the opinionated sentences for scoring features.

Error analysis revealed that a large number of the problems we experience are due to the fact that features can be expressed in text using different expressions. For this reasons more refined methods for identifying occurrences of the features in a text should be explored, including the use of coreference resolution. The weighting method currently used relies on frequency, however other methods for counting features should be investigated as well.

Finally, the motivation for this research is to optimise the dialogue between a user and an IQA system for selecting mobile phones. In light to this, the best way to prove the usefulness of the ranking methods is to carry out an extrinsic evaluation. This type of evaluation will be considered in the future.

References

1. Hervé Abdi. Kendall rank correlation. In N.J. Salkind, editor, *Encyclopedia of Measurement and Statistics*, pages 508–510. Thousand Oaks (CA): Sage, 2007.
2. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
3. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.
4. ZHU Junwu, LI Bin, WANG Fei, and WANG Sicheng. Mobile ontology. *JDCTA: International Journal of Digital Content Technology and its Applications*, 4(5):46–54, 2010.
5. H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165, 1958.
6. J.S. Maritz. *Distribution-free statistical methods*. Science Paperbacks. Chapman and Hall, 1984.
7. Xinfan Meng and Houfeng Wang. Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 177–180, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
8. Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
9. Yan Qu and Nancy Green. A constraint-based approach for cooperative information-seeking dialogue. In *Proceedings of the Second International Natural Language Generation Conference*, 2002.
10. Santosh Raju, Prasad Pingali, and Vasudeva Varma. An unsupervised approach to product attribute extraction. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 796–800, Berlin, Heidelberg, 2009. Springer-Verlag.

11. Verena Rieser and Oliver Lemon. Does this list contain what you were searching for? Learning adaptive dialogue strategies for interactive question answering. *Natural Language Engineering*, 15(1):55–72, January 2009.
12. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, pages 1–41, 2011.
13. Maite Taboada, Caroline Anthony, and Kimberly Voll. Methods for creating semantic orientation dictionaries. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, pages 427–432, Genoa, Italy, May 2006.
14. S. Varges, F. Weng, and H. Pon-Barry. Interactive question answering and constraint relaxation in spoken dialogue systems. *Natural Language Engineering*, 15(1):9–30, 2007.
15. Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 347–354, New York, NY, USA, 2011. ACM.
16. Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Constrained lda for grouping product features in opinion mining. In *Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part I*, PAKDD'11, pages 448–459, Berlin, Heidelberg, 2011. Springer-Verlag.