Structural Normalisation Methods for Improving Best Answer Identification in Question Answering Communities

Grégoire Burel, Paul Mulholland and Harith Alani
Knowledge Media Institute, Open University, UK
{g.burel, p.mulholland, h.alani}@open.ac.uk

ABSTRACT

Nowadays, Question Answering (Q&A) websites are popular source of information for finding answers to all kind of questions. Due to this popularity it is critical to help the identification of best answers to existing questions for simplifying the access to relevant information.

Although it is possible to identify relatively accurately *best answers* by using binary classifiers coupled with *user*, *content* and *thread* features, existing works have generally ignored to incorporate the thread-like structure of Q&A communities in the design of best answer identification predictors.

This paper investigates this particular issue by studying structural normalisation techniques for improving the accuracy of feature based *best answer* identification models.

Thread-based normalisation methods are introduced for improving the accuracy of identification models by introducing a systematic normalisation approach that normalise predictors by taking into account relations between features and the thread-like structure of Q&A communities.

Compared to similar non normalised models, better results are obtained for each of the three communities studied. These results show that structural normalisation methods can improve the identification of best answers compared to non-normalised models.

Keywords

Social Q&A platforms; online communities; best answers identification; structural normalisation; social media.

1. INTRODUCTION

With the general increase of interest in online Question Answering (Q&A) communities, it has become critical for the users of such services to easily identify the answers they are looking for. Unfortunately with the large amount of questions posted each day, many question threads lack information about the quality of answers. In this context the automatic identification of best answers may help theses communities to work more efficiently.

Although previous research shows that best answers can be identified relatively accurately using standard binary classifiers,[1, 2, 4, 3, 10, 7, 9] existing work have generally ignored the thread-like structure of Q&A communities when designing feature normalisation methods and best answer identification models.

This paper explores if structural normalisation techniques can improve existing models by investigating if *the thread-like structure of* Q&A *communities can help the automatic identification of best answers*.

In previous works, it has been shown that *thread* features are useful as they present relations between answers of a same answering thread [4]. Similarly, other works such as Gkotsis et al. [7] on the usage of normalised shallow features demonstrated that taking into account feature value order between answers of a same thread improved best answer identification. Building on those previous contributions, this paper propose to generalise and extend the concepts of thread features [4] and the ranking method proposed by Gkotsis et al.[7]. Consequently, the main contributions of this paper are: 1) Introduce a systematic approach for normalising features based on answering threads; 2) Compare the applicability of three different thread based normalisation methods: min-max normalisation, order normalisation and normalised order normalisation; 3) Investigate the impact of rank based features on best answers binary classifiers, and; 4) Investigate if structural normalisation improves best answer identification.

2. RELATED WORK

Many works have directly investigated the identification of *best answers*.[1, 2, 4, 3, 10, 7, 9]. Most of such works have used feature based models and decision tree based models in order to identify *best answers* with generally good accuracy by developing different types of features such as: 1) User features that represent the characteristics of authors of questions and answers; 2) Content features that represent the attributes of questions and answers, and; 3) Thread features that represent relations between answers in a particular answering thread [4].

The idea of using relational features between answers of a same thread or questions and answers has been studied in some recent work [10, 4, 7] but the scale of such analysis has been limited to specific features and not studied systematically across all the used features. For example Qiongjie et al.[10] used cosine similarity metrics between answers and question and answers and found that the minimum similarity between answers as well as the number of concurrent answers helps the identification of *best answers*. Similarly, our previous work [4] developed a few thread features by relating them according to their relative value (i.e. ratios) within an answering thread. The result generally showed that such type of feature where the best performers for identifying best answers. However, only a few features were specially designed and the approach was not generalised. Gkotsis et al. [7] used a slightly different approach and normalised textual features using their ranking within a thread

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890570 .

(e.g. they replaced the answer length features by a discrete number corresponding to their relative length within a thread.). Although their analysis was performed on multiple datasets from the STACK EXCHANGE (SE) websites and they obtained good results, their approach was restricted to predetermined features and not generalised.

In the area of feature normalisation, existing works have generally not used any specific normalisation techniques. For example some work on answer quality by Jeon et al.[8] used KERNEL DENSITY ESTIMATION (KDE) for improving the association of existing quality answers to new questions. Although the proposed approach displayed improvement compared to non-normalised approaches, this method did not take into account the structure of Q&A communities and was not a type of structural normalisation method. This method was also not used in the context of *best answer* identification.

Our work differs from previous research as we focus on evaluating if feature-based *best answer* identification models can be improved by using different thread-based normalisation methods. The presented work both generalise our previous work on thread features [4] and Gkotsis et al. research [7]. Besides generalising, formalising and comparing the impact of structural feature normalisation on *best answer* identification this paper also study how and why features importance changes when structural normalisation is applied.

Table 1: List of features and features categories.

	Features Set	
Type	Core Features Set (28)	Extended Features Set (30)
User	*Reputation, Community Age, Post Rate, Asking Rate, Answering Rate, Normalised Activity Entropy, Number of Posts, Number of Answers, Answers Ratio, Number of Best Answers, Best Answers Ratio, Number of Questions, Questions Ratio, Normalised Topic Entropy, Topical Reputation, Z-score, Question Success, Question Success Ratio.* (18)	*Reputation, Community Age, Post Rate, Asking Rate, Answering Rate, Normalised Activity Entropy, Number of Posts, Number of Answers, Answers Ratio, Number of Best Answers, Best Answers Ratio, Number of Questions, Questions Ratio, Normalised Topic Entropy, Topical Reputation, Z-score, Question Success, Question Success Ratio.* (18)
Content	*Answer Age, Number of Question Views, Number of Words, Gunning Fog Index, Flesch-Kinkaid Grade Level, Term Entropy.* (6)	***Number of Comments**, Answer Age, Number of Question Views, Number of Words, Gunning Fog Index, Flesch-Kinkaid Grade Level, Term Entropy.* (7)
Thread	*Number of Answers, Answer Position, Relative Answer Position, Topical Reputation Ratio.* (4)	***Score Ratio**, Number of Answers, Answer Position, Relative Answer Position, Topical Reputation Ratio.* (5)

3. FEATURE-BASED BEST ANSWERS IDENTIFICATION

As previously observed, many models are based on features-based binary classifiers that use a list of features for identifying if a given answer is a *best answer* therefore, we decide to assess the proposed structural methods against a reference feature-based best answer identification model.

3.1 Reference Model

In this paper, we decide to use an identification model that is based on our previous work on best answer identification [4] and Q&A communities [5, 6]. The model uses an *Alternating Decision Tree* algorithm and a set of 30 features divided into three different categories: 1) User features group predictors that describe the characteristic and reputation of authors of questions and answers; 2) Content features group the attributes of questions and answers, and; 3) Thread features contains a small set of predictors that encode relation between answers in a particular thread.

The name of each used features are reproduced in Table 1. Due to a lack of space we do not reproduce the full explanation of each features as their respective description can be found in our previous works [4, 5, 6].

3.2 Datasets

The proposed analysis is conducted on three different datasets. The first two are subs communities extracted from the April 2011 SE public datasets:[1] the SF user group and the non technical CO website composed of cooking enthusiasts. The other dataset is obtained from the SAP COMMUNITY NETWORK (SCN) forums and consists of posts submitted between December 2003 and July 2011.

The SCN forum dataset consists of $95,015$ threads and $427,221$ posts divided between $32,942$ users collected from 33 different forums between December 2003 and July 2011. Within those threads, we select threads that have best answers. Our final dataset consists of $29,960$ (32%) questions and $111,719$ (26%) answers.

[1]As part of the public SE dataset, the SERVER FAULT (SF) and COOKING (CO) datasets are available online at `http://www.clearbits.net/get/1698-apr-2011.torrent`.

The SF dataset has $71,962$ questions, $162,401$ answers and $51,727$ users. Within those questions we selected only the questions that have best answers. The final SF dataset has $36,717$ (51%) questions and $95,367$ (59%) answers.

The CO dataset has $3,065$ questions, $9,820$ answers and $4,941$ users. Similarly to the other datasets, we only select the questions that have best answers. The final dataset is composed of $2,154$ (70%) questions and $7,039$ (72%) answers.

4. THREAD-WISE OPTIMISATIONS FOR PREDICTING BEST ANSWERS

The structure of the Q&A communities analysed in this paper is centred around the concept of answering threads where each question is associated with a set of answers and where only a particular answer can be considered a best answer. As previous research has found, thread features are highly associated with best answers [4]. Such observation can be exploited to generalise thread features to all the predictors used in best answers identification models.

4.1 Thread-wise Normalisation

Feature normalisation has been used in different MACHINE LEARNING (ML) settings in order to deal with features that have outliers and ensure that ML algorithms consider independent features equally during the learning and prediction phases. A typical approach used for normalising features is based on the min/max formula[2] that scale numerical variables between 0 and 1. Unfortunately, such approach requires the knowledge of the boundaries of the studied variable which may be shift when additional data is analysed. For example, in Q&A communities, the *reputation* of users has no boundaries therefore min/max normalisation is not easily applicable. Another issue is the use of global minima and maxima instead of their local

[2]The min/max normalisation function $MM(x, X)$ that returns a normalised value of a given feature value $x \in X$, where X is the observed set of all the values of a particular feature is given by:

$$MM(x, X) = \frac{x - \min X}{\max X - \min X}$$

counterparts (i.e. community extrema instead of answering threads extrema). As previous work has highlighted, the usage of taking into account the relative values of a given feature within a thread helps the identification of best answers (i.e. the local relations for a given features are more useful than community wide values).

Calculating features ratios such as *score ratios* improve the ability to identify best answers compared to the scores of individual answers [4]. Following this observation different normalisation methods can be extrapolated. As a consequence all features become thread features as they represent the comparison of predictors values across threads.

In the following section, different normalisation schemes are proposed. In particular, the localised min-max approach is proposed and the ordering approach used by Gkotsis et al.[7] is generalised and extended by normalising the orders across question threads.

4.1.1 Min-Max Normalisation

The min-max feature normalisation approach is a localised version of the min-max. Consequently, each feature value within a particular thread is normalised based on the maximum and minimum of that feature for this particular thread. Formally, the thread normalisation function $TN_{minmax}(v_i, V_{f,t})$ normalise a value v_i of a given feature $f \in F$ within a question thread $t \in T$ where $v_i \in V_{f,t}$ and $V_{f,t}$ contains all the values of f for the thread t. As a result, the thread ratio normalisation function $TN_{minmax}(v_i, V_{f,t})$ is defined as follow:

$$TN_{minmax}(v_i, V_{f,t}) = \frac{v_i - \min V_{f,t}}{\max V_{f,t} - \min V_{f,t}} \quad (1)$$

For example for an answering thread with a feature that takes the values $V = \{31, 10, 5\}$, the corresponding normalised values are $V_{minmax} = \{1, 0.19, 0\}$.

4.1.2 Order Normalisation

The order normalisation approach generalises the approach presented by Gkotsis et al.[7] to any feature. Each feature is given a rank between 1 and the length of a question thread. If the value is the smallest for a given feature in a thread, it is given a value of one. If it is the highest value, it is given a value that equals the length of the thread. Intermediate values are valued similarly.

Using the same notation as the proportional normalisation method, the order normalisation function $TN_{order}(v_i, V_{f,t})$ is designed to return the index of a given value v_i in a given list of values ordered by decreasing order $V_{f,t}$. The returned value is bounded according to $\|V_{f,t}\|$ (i.e. $[1, \|V_{f,t}\|]$).

For example for an answering thread with a feature that takes the values $V = \{31, 10, 5\}$, the corresponding normalised values are $V_{order} = \{1, 2, 3\}$.

4.1.3 Normalised Order Normalisation

The normalised order method is based on the previous order normalisation approach. However, instead of returning absolute numbers, it divide the results by the length of the thread so that across threads, the normalisation is always bounded between zero and one. Given the order normalisation function $TN_{order}(v_i, V_{f,t})$, the normalised order function is given by:

$$TN_{orat}(v_i, V_{f,t}) = \frac{TN_{order}(v_i, V_{f,t})}{\|V_{f,t}\|} \quad (2)$$

For instance for an answering thread with a feature that takes the values $V = \{31, 10, 5\}$, the corresponding normalised values are $V_{order} = \{\frac{1}{3}, \frac{2}{3}, \frac{3}{3}\}$.

Table 2: Average IG for each dataset and different thread normalisation approach for identifying *best answers*.

		Normalisation Method		
Dataset	Original	Min-Max	Order	Norm. Order
SCN	0.0642	0.0721	0.1105	0.0891
Server Fault	0.0476	0.0330	0.1387	0.1044
Cooking	0.0654	0.0485	0.1137	0.0620
Average	0.0591	0.0407	0.1210	0.0851

4.1.4 Adaptive Features Normalisation

Some features do not necessarily vary within threads such as the *number of answers* or the *number of question views* therefore, normalising them will not be useful as such predictors only vary across threads. In order to account for such type of features automatically, the variance of values within threads for the whole dataset for a given feature is calculated. If the variance is zero and remains constant between all the threads, the features is not normalised. Otherwise, the feature is normalised with one of the previous functions.

5. NORMALISATION METHOD SELECTION

Before evaluating how thread normalisation impacts the identification of *best answers*, it is important to determine what normalisation approach is the most likely to provide the best results. In order to find the approach that works best for the three datasets studied, the average Information Gain (IG) for *best answer* identification of all the features presented in section 3 is compared for each dataset and with the three normalisation methods (Table 2).

The average IG of the normalised feature generally shows an increase compared with the non normalised features except for the min-max method. In particular, the order normalisation approach provides the highest gains with an average IG of 0.1210. The normalised order normalisation also provides good results with an IG of 0.0851 whereas the min-max approach does not improve IG (0.0407).

Those results show that in general normalisation approaches can improve *best answer* identification compared with the absence of normalisation. The order method seems to provide the best result, therefore it is retained as the normalisation approach applied in the rest of this paper.

6. BEST ANSWERS IDENTIFICATION USING THREAD-WISE NORMALISATION

Although good prediction results may be obtained when using the reference model and the features described in section 3, such type of model is not optimised using any structural normalisation approaches even though it can be expected that normalisation methods such as feature normalisation can increase the accuracy of ML tasks.

The following experiments aim at evaluating the impact of thread normalisation on *best answer* identification by highlighting how each feature impacts prediction accuracy for each of our datasets. The goal is to determine if *structural normalisation improves automatic* best answer *identification*.

6.1 Experimental Setting

In this experiment, the impact of order normalisation on *best answer* identification is compared for each of the three studied datasets. Each normalisation method is applied on the reference model and each of the features sets described in section 3.

In order to evaluate our result, a 10-folds stratified cross-validation is performed. The precision (P), recall (R) and the harmonic mean F-measure (F_1) are reported as well as the area under the Receiver Operator Curve (ROC) measure. The experiment is done using the *Alternating Decision Tree* algorithm and the normalised and non normalised results are compared. The features that are the most relevant are also discussed by reporting the INFORMATION GAIN RATIO (IGR) of individual features.

6.2 Results: Model Comparison

For comparing the impact of thread normalisation with the non-normalised features, the results for both the normalised and non-normalised results are reported. The results are listed in Table 3.

6.2.1 Baseline Models:

The normalisation approach shows a relatively good performance of the *number of words* feature. For the non-normalised features, on average, $F_1 = 0.526$ and for the order normalised version, on average $F_1 = 0.718$ ($+26.8\%$). This results shows that the length of answers can identify *best answers* when the relative length of answers is used.

Similarly to previous observations [4], the *answer score* and *answer score ratios* features are very good predictors of *best answers*. In particular, by using thread normalisation, both features become very good predictors with an average F_1 of 0.839.

Looking at the distribution of baseline normalised features, it can be observed that answers that are longer than the other thread answers are more likely to be *best answers*. Similarly higher score means better answers. Such results are again similar to past results [4].

Overall the normalisation approach benefits a lot the *answer score* and *answer score ratios* features. This observation confirms that relational features (i.e. thread features) and *score* based metrics are very good *best answer* predictors.

6.2.2 Core Features Models:

Lets now focus on the core feature types (i.e. *users*, *content* and *threads*) for analysing the impact of feature sets on the identification process. Similarly to the *baseline* features, higher precision/recall compared to the non normalised models is found.

For the SCN and SF communities and the non-normalised features, the least useful features are content feature (median $F_1 = 0.614$) followed by the user features (median $F_1 = 0.615$) and thread features (median $F_1 = 0.733$).

Although a general increase in F_1 appears compared to the non-normalised features, the impact of feature set is largely different as all features become relational. In this situation, the *thread* features are the least efficient with a median $F_1 = 0.73$ (with no observable real difference compared with the non-normalised features) followed by the *user* features ($F_1 = 0.74$, $+20.3\%$ compared with the non-normalised features) and the *content* features ($F_1 = 0.744$, $+21\%$ compared with the non-normalised features).

Since all features become *thread* features it is somehow expected that they perform lower than the other feature sets as the *thread* set has significantly less features than the other sets. Even thought, the difference between F_1 medians of the *users* and *content* feature set is minimal ($< 1\%$), it appears that content features play a higher role when relations between answers are taken into account. This result confirms the findings of Gkotsis et al. [7] that shallow content features are efficient for distinguishing quality and low quality answer within threads. These findings also highlight that reputation information about user may be only useful when used globally (i.e. distinguishing quality answers at the community level) rather than locally (i.e. distinguishing quality answers at the thread level).

Using all non-normalised features give better result than only relying on individual feature sets. Such results are similar to previous research [4]. When the order normalisation is used, results highlight similar patterns with *score ratios* giving high accuracy. In general, it appears that the *all* normalised feature perform better than the *all* non-normalised feature set with a respective average F_1 of 0.762 and 0.752.

6.2.3 Extended Features Models:

The main difference between using *core* and *extended* features is the presence of *scores*. The presence of such scores makes evident the importance of scores as *content* and *thread* feature become the best feature sets compared to the *user* set when using normalised features (Table 3). When not using normalised features the results are again similar to previous work where *thread* features produced better performances than the other sets [4].

Looking at the combined feature sets (Table 3), it appears that the results are not significantly different when using or not using thread normalisation with a median F_1 of 0.834 when normal features are used and 0.84 when order normalisation is applied. However, the thread normalisation approach gives better precision with a median precision of 0.839 instead of 0.822 ($+2\%$).

As a summary, it appears that thread normalisation approaches improves *best answer* identification. A tailed pared t-test between the results of the non-normalised models and the normalised models for each datasets and features sets confirms such relation with a $p-$value of $2.817e-05$. On average, an increase in F_1 of $+5.3\%$ is observed compared to the non normalised models.

6.3 Results: Feature Selection

Following the last experiment, the second analysis evaluates the importance of individual features based on their order normalisation. In order to infer what normalised features are the most important, the IGR of the top features is calculated for each of our datasets and for the non-normalised and normalised methods in Table 4.

6.3.1 Core Features:

First, the focus is on the *core feature* set. Table 4 shows that SCN's most important feature appears to be the *ratio of answers* posted by answers authors. Such feature seems not as important for the other datasets (ranked $12th$ for SF and $> 15th$ for CO). The *user reputation* feature seems important for each dataset (ranked $3rd$ for SCN, $9th$ for SF and $8th$ for CO) meaning that the amount of knowledge users have may influence *best answer* identification positively. For SCN, *best answers* are correlated with the most knowledgeable users (i.e. higher reputation). The *term entropy* feature is generally well ranked (ranked $10th$ for SCN and $5th$ for SF and CO). For SF and CO, it appears that the answer that have more diverse vocabulary are more likely to be *best answers*. This shows that *best answers* may be more detailed compared to the other answers of the same thread.

Compared to the ranking of the non-normalised features, *user* features also play a dominant role. However, *topic reputation* does not seem to be an important feature in this context, meaning that this feature only helps when distinguishing *best answers* in a global context. In opposition, the user tendency to answer questions becomes useful when used for distinguishing *best answers* within threads as users that are focused on answering seem to provide better answers.

Table 3: Average answer *Precision*, *Recall*, F_1 and AUC for the *SCN Forums*, *Server Fault* and *Cooking* datasets for different feature sets and extended features sets (marked with +) and reduced features sets (marked with -) using the *Alternating Decision Tree* classifier and thread order normalisation.

		SCN Forums				Server Fault				Cooking			
Model	Features	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1	AUC
Std.	Words	0.500	0.360	0.419	0.611	0.519	0.590	0.552	0.566	0.566	0.651	0.606	0.652
	Answer Score	-	-	-	-	0.592	0.635	0.613	0.672	0.692	0.719	0.705	0.795
	Answer Sc. Ratio	-	-	-	-	0.783	0.801	0.792	0.847	0.824	0.839	0.831	0.908
	Users	0.565	0.674	0.615	0.755	0.593	0.632	0.612	0.669	0.592	0.661	0.624	0.687
	Content	0.550	0.656	0.599	0.673	0.592	0.637	0.614	0.674	0.625	0.687	0.654	0.737
	Threads	0.727	0.788	0.756	0.860	0.720	0.745	0.733	0.807	0.653	0.773	0.708	0.783
	All	0.753	0.811	0.781	0.883	0.725	0.777	0.750	0.829	0.687	0.764	0.724	0.817
	Users+	-	-	-	-	0.593	0.632	0.612	0.669	0.592	0.661	0.624	0.687
	Content+	-	-	-	-	0.681	0.692	0.686	0.761	0.734	0.755	0.744	0.843
	Threads+	-	-	-	-	0.820	0.842	0.831	0.908	0.828	0.854	0.841	0.912
	All+	-	-	-	-	0.823	0.844	0.833	0.912	0.821	0.849	0.835	0.913
Norm.	Words	0.713	0.713	0.713	0.763	0.727	0.727	0.727	0.771	0.715	0.715	0.715	0.765
	Answer Score	-	-	-	-	0.826	0.826	0.826	0.863	0.853	0.853	0.853	0.884
	Answer Sc. Ratio	-	-	-	-	0.826	0.826	0.826	0.863	0.853	0.853	0.853	0.884
	Users	0.725	0.799	0.760	0.855	0.717	0.765	0.740	0.811	0.682	0.761	0.719	0.791
	Content	0.701	0.792	0.744	0.796	0.727	0.763	0.744	0.815	0.681	0.738	0.708	0.790
	Threads	0.665	0.847	0.745	0.819	0.721	0.739	0.730	0.804	0.650	0.761	0.701	0.771
	All	0.772	0.807	0.789	0.877	0.731	0.778	0.754	0.833	0.723	0.766	0.744	0.824
	Users+	-	-	-	-	0.717	0.765	0.740	0.811	0.682	0.761	0.719	0.791
	Content+	-	-	-	-	0.824	0.829	0.826	0.901	0.847	0.855	0.851	0.913
	Threads+	-	-	-	-	0.826	0.826	0.826	0.903	0.853	0.853	0.853	0.903
	All+	-	-	-	-	0.831	0.828	0.829	0.910	0.848	0.855	0.851	0.914

6.3.2 Extended Features:

When observing extended features, the score measures are the most important ($+40\%$ IGR and $+54\%$ IGR for SF and CO compared to the second ranked features). Both *score ratios* and *scores* are ranked at the same position as both methods metrics become the same when normalised. Such results are largely comparable to the non-normalised features where *score ratios* is ranked the highest.

Compared with the non-normalised features rankings, the *number of comments*, which only exists in the CO and SF datasets, appear important as high comments correlate with good answers. For example users may use comment sections to thank users for a good answer. Therefore, the relative amount of comments may be a good indicator of *best answers*.

As a summary, a difference between non-normalised and normalised rankings can be observed. Although, ratings remain highly correlated with quality content in each case, it seems that content features are more important when used as relations rather than when used globally. This shows that the impact of features is highly different when used locally (i.e. when comparing within a thread) compared to globally (i.e. when comparing across all the answer of a community).

7. DISCUSSION

In order to improve existing classification models, different methods based on the hypothesis that the *thread-like structure of* Q&A *communities can help the automatic identification of* best answers were explored. Although this work is similar to previous research,[7] this contribution varies significantly as the concept of thread normalisation was formalised and different normalisation techniques were introduced. In addition we also introduced the idea of adaptive normalisation, a method for automatically identifying what features need to be normalised.

The results show differences in accuracy when using non-normalised features and when using relational features (i.e. thread normalisation). The thread normalisation showed that content features are good locally (i.e. at the tread level) even though they are not useful when used globally. This result shows the importance of normalisation as features with limited utility become relevant thanks to simple transformations

In general, the usage of thread-wise normalisation techniques proved to improve results compared to their non-normalised counterparts, therefore, the structural normalisation methods proposed in this paper appear to improve *best answer* prediction in each of the dataset we studied. As a result, it can be argued that structural normalisation helps *best answer* identification.

8. CONCLUSION

Feature based models have proven to be a good method for identifying *best answers*. In this paper, different approaches for improving such models were proposed based on the hypothesis that *the thread-like structure of* Q&A *communities can help the automatic identification of* best answers.

Based on IG analysis, order normalisation appeared to be the most useful normalisation technique. Although on average across all the features sets only an improvement of $+5.3\%$ was reported compared to the usage of non-normalised features, this improvement is consistent across all the datasets and significant ($p = 0.00002817$) even thought the improvement over the previous best result is not important.

The normalisation method highlighted the importance of content features when used at the thread level such as *term entropy*. This observation shows that some features become only useful when used as relations.

Acknowledgement: This work is partly funded by the EC-FP7 project DecarboNet (grant number 265454).

Table 4: Top order non-normalised and normalised features ranked by Information Gain Ratio for the *SCN*, *Server Fault* and *Cooking* datasets. Type of feature is indicated by U/C/T for User/Content/Thread.

		SCN		Server Fault		Cooking	
Norm. Type	R.	IGR	Feature	IGR	Feature	IGR	Feature
None	1	0.0832	*Topic Rep. Ratio (T)*	0.1016	***Score Ratio (T)***	0.1552	***Score Ratio (T)***
	2	0.0588	*Nb. Answers (T)*	0.0914	*Nb. Answers (T)*	0.0833	*Topic Rep. Ratio (T)*
	3	0.0478	*Topic Rep. (U)*	0.0553	*Topic Rep. Ratio (T)*	0.0702	***Score (C)***
	4	0.0368	*A. Succ. Ratio (U)*	0.0518	*Position (T)*	0.0619	*Nb. Answers (T)*
	5	0.0337	*Reputation (U)*	0.0305	***Score (C)***	0.0535	*Position (T)*
	6	0.0327	*Activity Entropy (U)*	0.0296	*Rel. Position (T)*	0.0446	*Answer Age (C)*
	7	0.0317	*Nb. Bests (U)*	0.0223	*Answer Age (C)*	0.0354	*Nb. Bests (U)*
	8	0.0316	*Question Ratio (U)*	0.0193	***Nb. Comments (C)***	0.0332	*Reputation (U)*
	9	0.0312	*Answer Ratio (U)*	0.0161	*Q. Views (C)*	0.0315	***Nb. Comments (C)***
	10	0.0278	*Rel. Position (T)*	0.0140	*A. Succ. Ratio (U)*	0.0313	*Post Rate (U)*
	11	0.0277	*Z-Score (U)*	0.0090	*Z-Score (U)*	0.0307	*A. Succ. Ratio (U)*
	12	0.0229	*Position (T)*	0.0088	*Nb. Posts (U)*	0.0269	*Nb. Posts (U)*
	13	0.0152	*Nb. Answers (U)*	0.0081	*Community Age (U)*	0.0257	*Topic Entropy (U)*
	14	0.0150	*Asking Rate (U)*	0.0078	*Reputation (U)*	0.0250	*Z-Score (U)*
	15	0.0123	*Nb. Posts (U)*	0.0073	*Answering Rate (U)*	0.0243	*Term Entropy (C)*
Order Norm.	1	0.0763	*Answer Ratio (U)*	0.1532	*Score (C)*	0.1732	*Score (C)*
	2	0.0747	*Z-Score (U)*	0.1532	*Score Ratio (T)*	0.1732	*Score Ratio (T)*
	3	0.0683	*Reputation (U)*	0.0914	*Nb. Answers (T)*	0.0793	*Nb. Comments (C)*
	4	0.0681	*Nb. Answers (U)*	0.0809	*Nb. Comments (C)*	0.0686	*Nb. of Words (C)*
	5	0.0644	*Nb. Posts (U)*	0.0765	*Term Entropy (C)*	0.0674	*Term Entropy (C)*
	6	0.0607	*Nb. Bests (U)*	0.0754	*Nb. of Words (C)*	0.0651	*Nb. Bests (U)*
	7	0.0588	*Nb. Answers (T)*	0.0628	*A. Succ. Ratio (U)*	0.0650	*A. Succ. Ratio (U)*
	8	0.0546	*A. Succ. Ratio (U)*	0.0538	*Q. Succ. Ratio (U)*	0.0623	*Reputation (U)*
	9	0.0537	*Answering Rate (U)*	0.0527	*Reputation (U)*	0.0619	*Nb. Answers (T)*
	10	0.0536	*Term Entropy (C)*	0.0522	*Nb. Bests (U)*	0.0505	*Answering Rate (U)*
	11	0.0527	*Nb. of Words (C)*	0.0500	*Nb. Posts (U)*	0.0502	*Z-Score (U)*
	12	0.0510	*Community Age (U)*	0.0495	*Answer Ratio (U)*	0.0498	*Nb. Posts (U)*
	13	0.0474	*Topic Rep. (U)*	0.0482	*Nb. Answers (U)*	0.0497	*Nb. Solved (U)*
	14	0.0474	*Topic Rep. Ratio (T)*	0.0477	*Nb. Solved (U)*	0.0485	*Nb. Answers (U)*
	15	0.0466	*Post Rate (U)*	0.0476	*Question Ratio (U)*	0.0483	*Nb. Questions (U)*

9. REFERENCES

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and yahoo answers: Everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web*, volume 17 of *WWW '08*, pages 665–674, 2008.

[2] M. J. Blooma, A. Y. K. Chua, and D. H.-L. Goh. A predictive framework for retrieving the best answer. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1107–1111, 2008.

[3] M. J. Blooma, D. Hoe-Lian Goh, and A. Yeow-Kuan Chua. Predictors of high-quality answers. *Online Information Review*, 36(3):383–400, 2012.

[4] G. Burel, Y. He, and H. Alani. Automatic identification of best answers in online enquiry communities. In *Proceedings of the 9th international conference on The Semantic Web: research and applications*, pages 514–529. Springer-Verlag, 2012.

[5] G. Burel, P. Mulholland, Y. He, and H. Alani. Modelling question selection behaviour in online communities. In *Proceedings of the 24th International Conference on World Wide Web Companion*, WWW '15 Companion, pages 357–358. International World Wide Web Conferences Steering Committee, 2015.

[6] G. Burel, P. Mulholland, Y. He, and H. Alani. Predicting answering behaviour in online question answering communities. In *Proceedings of the 26th Conference on Hypertext and Social Media*, HT '15, 2015.

[7] G. Gkotsis, K. Stepanyan, C. Pedrinaci, J. Domingue, and M. Liakata. It's all in the content: State of the art best answer prediction based on discretisation of shallow linguistic features. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 202–210. ACM, 2014.

[8] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 228–235. ACM, 2006.

[9] C. Shah. Building a parsimonious model for identifying best answers using interaction history in community q&a. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, page 51. American Society for Information Science, 2015.

[10] Q. Tian, P. Zhang, and B. Li. Towards predicting the best answers in community-based question-answering services. In *ICWSM*, 2013.