Finding Question-Answer Pairs from Online Forums

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun

In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)

A classification-based approach to question answering in discussion boards

Liangjie Hong and Brian D. Davison

In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09)

Presented by Polykarpos Thomadakis

CS 834 - Introduction to Information Retrieval

Fall 2017

Old dominion university

Online Forum

- A web application for holding discussions and posting user generated content in a specific domain (sports, recreation, techniques etc.)

- Contains huge amount of valuable user generated content

- Highly desirable to extract this content and reuse it

Applications of Forum Mining

- Essential to many QA services
  - SE instant answers
  - QA search systems
  - Community-based Question Answering (CQA)

- A natural way to improve forum management
  - Query question-answer pairs extracted from forums

- Question-answer knowledge mined can be used to augment the knowledge base of chatbots

Challenge

- Each forum thread usually contains an initiating post and a couple of reply posts
- The initiating post usually contains several questions
- The asynchronous nature of forum discussion makes it common for multiple participants to pursue multiple questions in parallel
- Proposed method consists of two components
  - Question detection
  - Answer detection

Question Detection

- Harder than it looks:
  - Questions stated in informal way
  - In various formats
  - Question marks and 5W1H question words not adequate
    - 30% do not end with question marks
    - 9% ending with question marks are not questions (e.g. “really?”)

- Use labeled sequential patterns (LSPs) to build classifiers and complement the inadequacy of simple rules

Labeled Sequential Patterns (LSPs)

- A LSP , *p*, is an implication in the form of LHS —> c, where LHS is a sequence and c is a class label
- A sequence $s_1 =< a_1, \ldots, a_m >$ is contained in a sequence $s_2 =< b_1, \ldots, b_n >$ if
  - There exist integers $i_1 \ldots i_m$ such that $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ and $a_j = b_{i_j} \forall j \in 1, \ldots, m$
  - The distance between the two adjacent items $b_{i_j}$and $b_{i_{j+1}}$ in s_2 needs to be less than a threshold
- Similarly, LSP p_1 is contained by p_2 if the sequence $p_1.LHS$ is contained by $p_2.LHS$ and $p_1.c = p_2.c.$
- Support of LSP p -> the percentage of tuples in the collection containing p
- Confidence of LSP p -> Probability of p being true : $\frac{\sup(p)}{\sup(p.LHS)}$

Approach

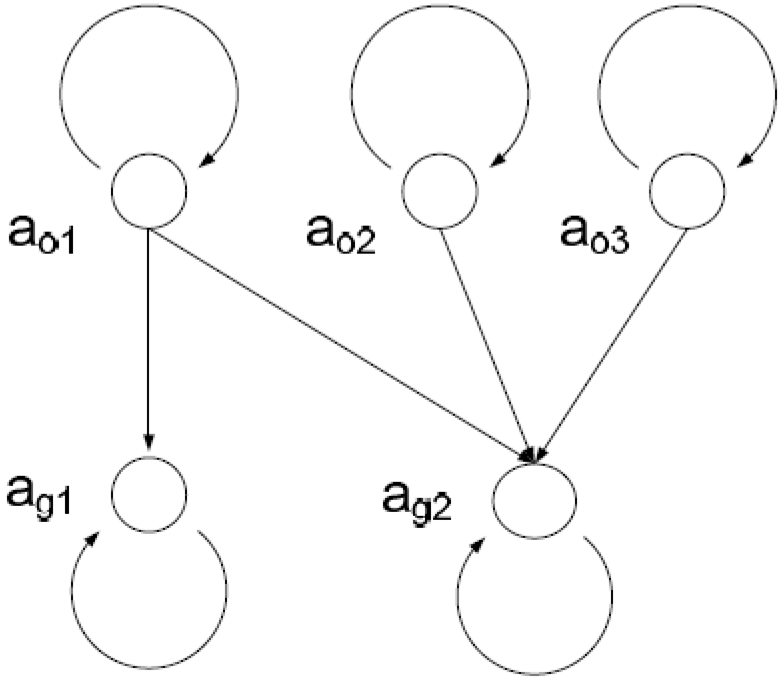
- Pre-process each sentence by applying Part-Of-Speech (POS) tagger
  - Keeping keywords including 5W1H and modal words
- Keywords are usually good indicators of questions
- POS can reduce the sparseness of words
- Mine LSPs by imposing minimum support and confidence threshold
  - Ensure that discovered patterns are general and are capable of predicting question or non-question sentences
- Build a classifier using each LSP as a feature
  - If a sentence includes a LSP the corresponding features is set to 1

Answer Detection

- Difficult because:
  - Multiple questions and answers may be discussed in parallel interweaved together
  - One post may contain answers to multiple questions
  - One question may have multiple replies
- Straightforward approach:
  - Cast answer-finding as a traditional document retrieval problem
  - Employ ranking methods (e.g. cosine similarity, query likelihood and KL-divergence LM)
  - Does not consider relationship of candidate answers and forum specific features
- Model relationship between candidate answers using a graph-based method
  - A candidate answer related to a candidate answer with high score, is also likely to be an answer even if it doesn't have a high score

Building Graphs

- Each candidate answer in A_q will correspond to a vertice V .
  - The problem is how to generate the edge set E.
- Given two candidate answers a_o and a_g
- If $\frac{1}{1 + KL\left(a_o \middle| a_g\right)} > \theta$, form an edge from a_o to a_g
  - a_g is a generator of a_o
  - a_o is an offspring of a_g



Computing Weights

- Use the KL-divergence score and augment it with two more factors:
- Distance:
  - Based on the observation that replying posts far away from the question are less likely to contain the answer d(a,q)
- Authority:
  - Answers from authors with high authority are more likely to contain answers
- Weight for edge $a_o \rightarrow a_g$: $\mathrm{w}(a_o \rightarrow a_g) = \frac{1}{1+KL\left(P(a_o)\middle|P(a_g)\right)} + \frac{\lambda_1 1}{d(a_g,q)} + \lambda_2 author(a_g)$
- The result is normalized across its generators

Score Propagation

- Propagation without initial score:
  - Compute initial ranking scores using cosine similarity, query likelihood or KL-divergence LM
  - Compute authority for each candidate answer

$$authority(a_g) = \sum_{a_o \in C_a} nw(a_o \rightarrow a_g) \times authority(a_o)$$

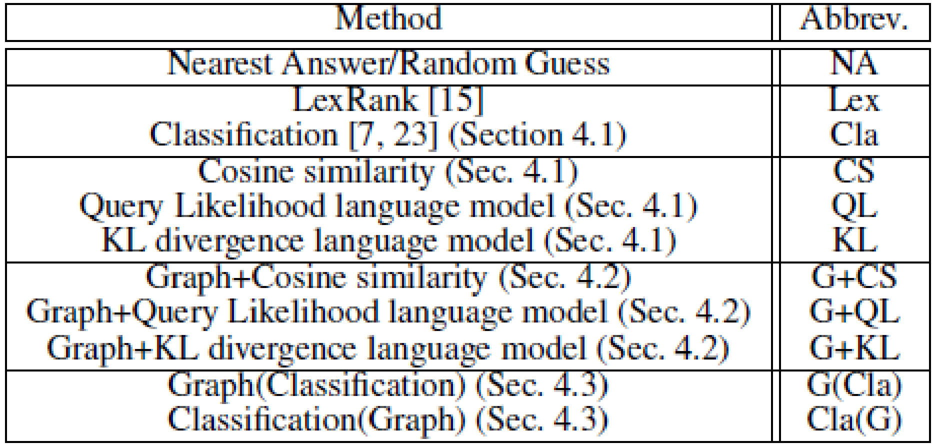
  - The product of authority and initial ranking score is the final score

$$Pr(\mathbf{q}|\mathbf{a}) := authority(\mathbf{a}) \times score(\mathbf{q}, \mathbf{a})$$

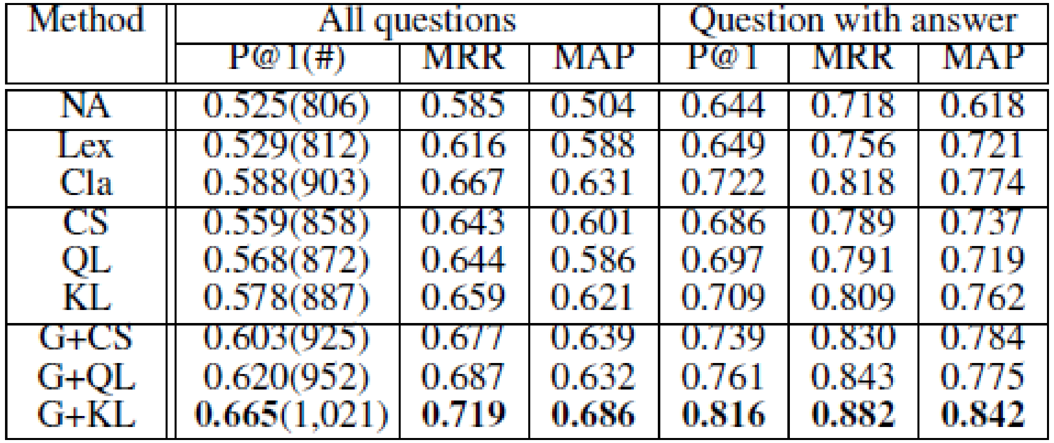
- Propagation with initial score:
  - Incorporates the initial score between candidate answer and question into propagation

$$Pr(\mathbf{q}|\mathbf{a}) = \lambda \frac{Pr(\mathbf{q}|\mathbf{a})}{\sum_{\mathbf{t} \in C_{\mathbf{q}}} Pr(\mathbf{q}|\mathbf{t})} + (1-\lambda) \sum_{\mathbf{v} \in C_{\mathbf{q}}} nw(\mathbf{v} \rightarrow \mathbf{a}) \times Pr(\mathbf{q}|\mathbf{v})$$

Results



Method	Abbrev.
Nearest Answer/Random Guess	NA
LexRank [15]	Lex
Classification [7, 23] (Section 4.1)	Cla
Cosine similarity (Sec. 4.1)	CS
Query Likelihood language model (Sec. 4.1)	QL
KL divergence language model (Sec. 4.1)	KL
Graph+Cosine similarity (Sec. 4.2)	G+CS
Graph+Query Likelihood language model (Sec. 4.2)	G+QL
Graph+KL divergence language model (Sec. 4.2)	G+KL
Graph(Classification) (Sec. 4.3)	G(Cla)
Classification(Graph) (Sec. 4.3)	Cla(G)



Method	All questions			Question with answer		
	P@1(#)	MRR	MAP	P@1	MRR	MAP
NA	0.525(806)	0.585	0.504	0.644	0.718	0.618
Lex	0.529(812)	0.616	0.588	0.649	0.756	0.721
Cla	0.588(903)	0.667	0.631	0.722	0.818	0.774
CS	0.559(858)	0.643	0.601	0.686	0.789	0.737
QL	0.568(872)	0.644	0.586	0.697	0.791	0.719
KL	0.578(887)	0.659	0.621	0.709	0.809	0.762
G+CS	0.603(925)	0.677	0.639	0.739	0.830	0.784
G+QL	0.620(952)	0.687	0.632	0.761	0.843	0.775
G+KL	**0.665**(1,021)	**0.719**	**0.686**	**0.816**	**0.882**	**0.842**

Finding Question-Answer Pairs from Online Forums

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun

In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)

A classification-based approach to question answering in discussion boards

Liangjie Hong and Brian D. Davison

In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09)

Same Goal

- Retrieve question and potential answers from forums automatically and effectively

- Detect question-related threads in an efficient manner

- Discover potential answers without analyzing the content of replies

- Can this task be treated as a traditional information retrieval problem?

Problem Definition

- A post is treated as a single question as a whole
    - Not focusing on question sentences or paragraphs

- Detect whether the first post is a “question post” containing at least one problem needed to be solved.

- Consider only the replies that answer directly the first post
    - Not the ones answering questions in other replies
    - Include replies providing links to other answers as answer posts

Question Detection

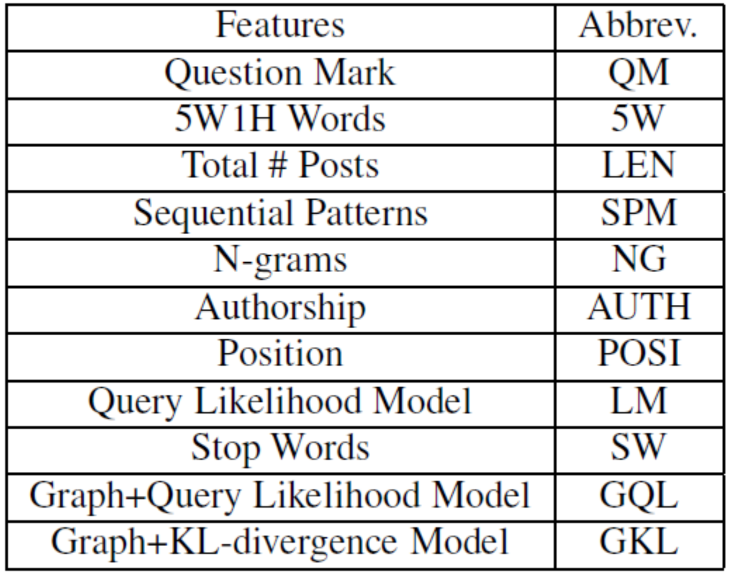
- Features used:
  - Question mark: Sentence ending with a question mark might be a question
  - 5W1H words: Probably used if the sentence is a question
  - Total number of posts: Empirically many replies indicate a question not well defined or that the topic shifts
  - Authorship: High quality content generated by highly authoritative authors. Users who usually answer other's questions
  - N-grams: Find word relations that are usually seen in questions

Answer Detection

Features used:

- The position of the answer post: The answer post usually appears not very close to the bottom
- Authorship: Same as before
- N-gram: Same as before
- Stop Words: See whether the answer contains more detailed and precise words rather than “stop words”
- Query Likelihood (Language Model): Calculate the likelihood that a replied post is relevant to the original question post

Results: Question Detection (Single features)



Features	Abbrev.
Question Mark	QM
5W1H Words	5W
Total # Posts	LEN
Sequential Patterns	SPM
N-grams	NG
Authorship	AUTH
Position	POSI
Query Likelihood Model	LM
Stop Words	SW
Graph+Query Likelihood Model	GQL
Graph+KL-divergence Model	GKL

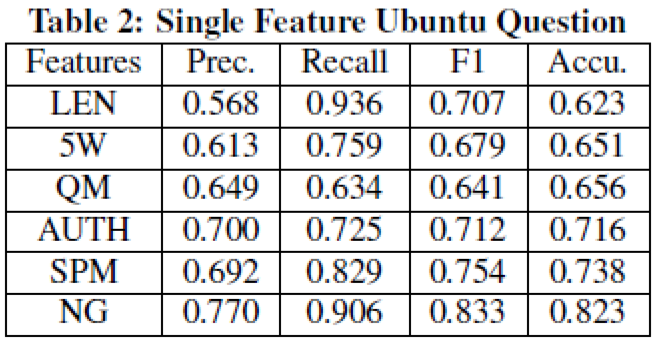


Table 2: Single Feature Ubuntu Question

Features	Prec.	Recall	F1	Accu.
LEN	0.568	0.936	0.707	0.623
5W	0.613	0.759	0.679	0.651
QM	0.649	0.634	0.641	0.656
AUTH	0.700	0.725	0.712	0.716
SPM	0.692	0.829	0.754	0.738
NG	0.770	0.906	0.833	0.823

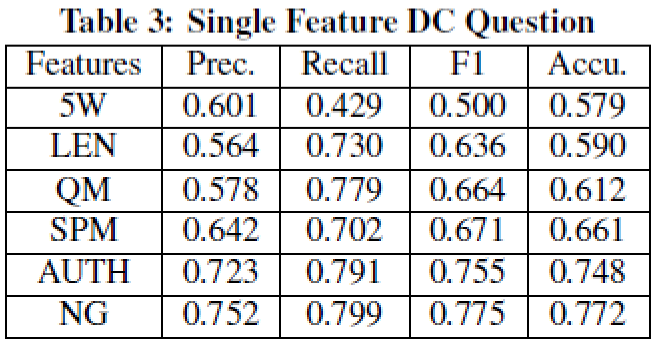


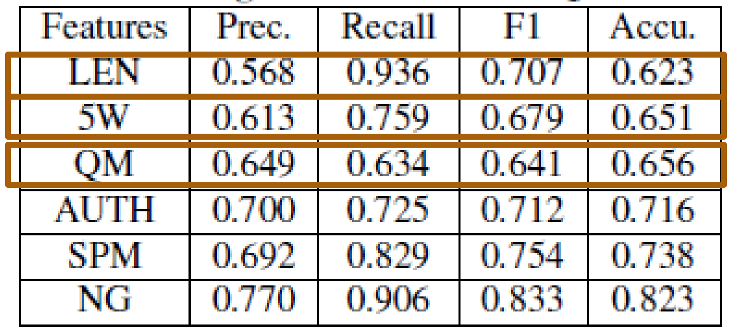
Table 3: Single Feature DC Question

Features	Prec.	Recall	F1	Accu.
5W	0.601	0.429	0.500	0.579
LEN	0.564	0.730	0.636	0.590
QM	0.578	0.779	0.664	0.612
SPM	0.642	0.702	0.671	0.661
AUTH	0.723	0.791	0.755	0.748
NG	0.752	0.799	0.775	0.772

N-grams perform better

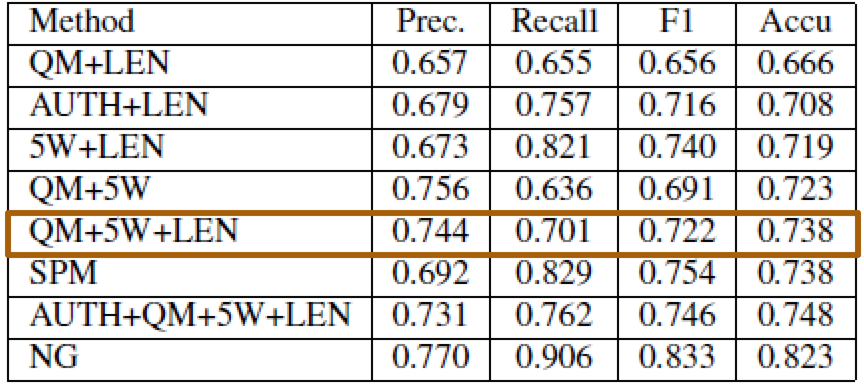
Results: Question Detection (Combined features)

Table 2: Single Feature Ubuntu Question



Features	Prec.	Recall	F1	Accu.
LEN	0.568	0.936	0.707	0.623
5W	0.613	0.759	0.679	0.651
QM	0.649	0.634	0.641	0.656
AUTH	0.700	0.725	0.712	0.716
SPM	0.692	0.829	0.754	0.738
NG	0.770	0.906	0.833	0.823

Table 5: Combined Features Ubuntu Question

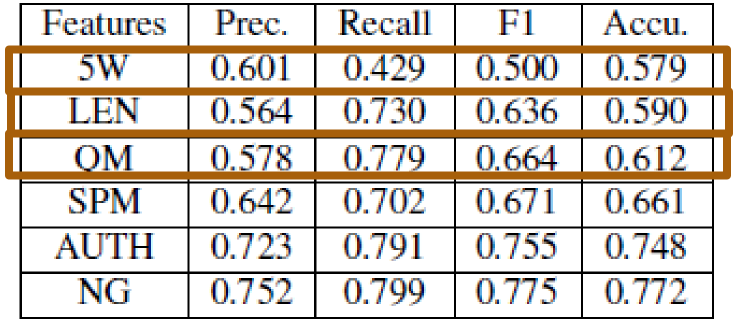


Method	Prec.	Recall	F1	Accu
QM+LEN	0.657	0.655	0.656	0.666
AUTH+LEN	0.679	0.757	0.716	0.708
5W+LEN	0.673	0.821	0.740	0.719
QM+5W	0.756	0.636	0.691	0.723
QM+5W+LEN	0.744	0.701	0.722	0.738
SPM	0.692	0.829	0.754	0.738
AUTH+QM+5W+LEN	0.731	0.762	0.746	0.748
NG	0.770	0.906	0.833	0.823

Need only local information

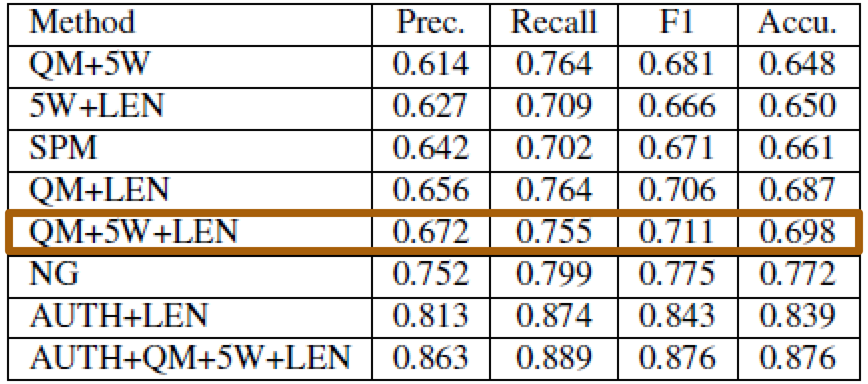
Combinations of features perform considerably better than single features

Table 3: Single Feature DC Question



Features	Prec.	Recall	F1	Accu.
5W	0.601	0.429	0.500	0.579
LEN	0.564	0.730	0.636	0.590
QM	0.578	0.779	0.664	0.612
SPM	0.642	0.702	0.671	0.661
AUTH	0.723	0.791	0.755	0.748
NG	0.752	0.799	0.775	0.772

Table 6: Combined Features DC Question



Method	Prec.	Recall	F1	Accu.
QM+5W	0.614	0.764	0.681	0.648
5W+LEN	0.627	0.709	0.666	0.650
SPM	0.642	0.702	0.671	0.661
QM+LEN	0.656	0.764	0.706	0.687
QM+5W+LEN	0.672	0.755	0.711	0.698
NG	0.752	0.799	0.775	0.772
AUTH+LEN	0.813	0.874	0.843	0.839
AUTH+QM+5W+LEN	0.863	0.889	0.876	0.876

Results: Answer Detection (Single features)

N-grams do not perform that well

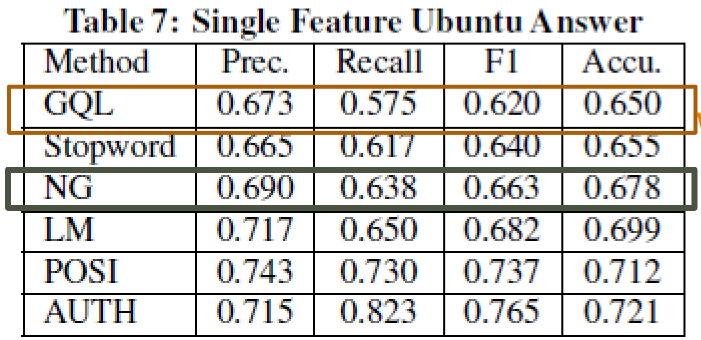


Table 7: Single Feature Ubuntu Answer

Method	Prec.	Recall	F1	Accu.
GQL	0.673	0.575	0.620	0.650
Stopword	0.665	0.617	0.640	0.655
NG	0.690	0.638	0.663	0.678
LM	0.717	0.650	0.682	0.699
POSI	0.743	0.730	0.737	0.712
AUTH	0.715	0.823	0.765	0.721

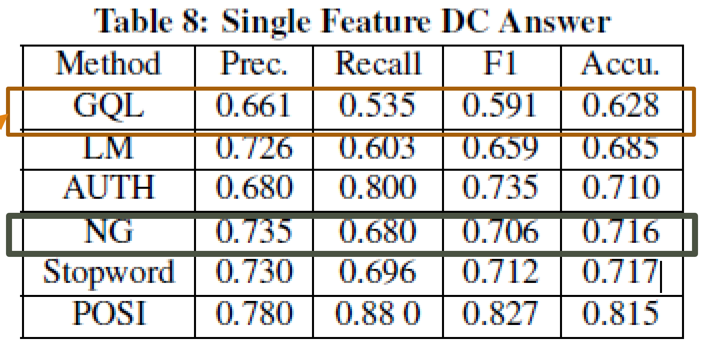


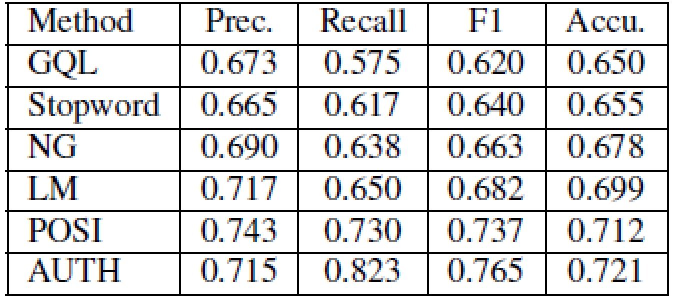
Table 8: Single Feature DC Answer

Method	Prec.	Recall	F1	Accu.
GQL	0.661	0.535	0.591	0.628
LM	0.726	0.603	0.659	0.685
AUTH	0.680	0.800	0.735	0.710
NG	0.735	0.680	0.706	0.716
Stopword	0.730	0.696	0.712	0.717
POSI	0.780	0.88 0	0.827	0.815

Performance using the method from the 1st paper

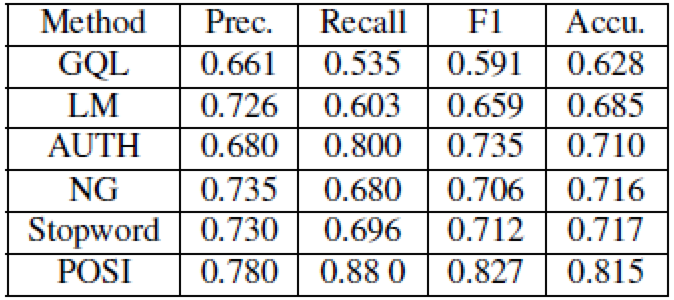
Results: Answer Detection (Combined features)

Table 7: Single Feature Ubuntu Answer



Method	Prec.	Recall	F1	Accu.
GQL	0.673	0.575	0.620	0.650
Stopword	0.665	0.617	0.640	0.655
NG	0.690	0.638	0.663	0.678
LM	0.717	0.650	0.682	0.699
POSI	0.743	0.730	0.737	0.712
AUTH	0.715	0.823	0.765	0.721

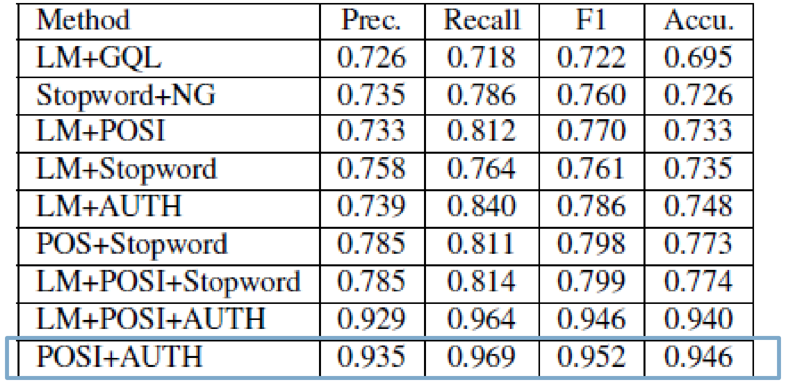
Table 8: Single Feature DC Answer



Method	Prec.	Recall	F1	Accu.
GQL	0.661	0.535	0.591	0.628
LM	0.726	0.603	0.659	0.685
AUTH	0.680	0.800	0.735	0.710
NG	0.735	0.680	0.706	0.716
Stopword	0.730	0.696	0.712	0.717
POSI	0.780	0.88 0	0.827	0.815

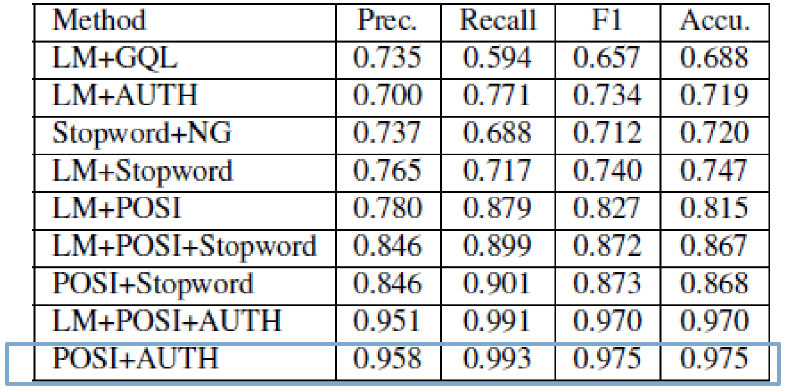
~ 30% improvement

Table 9: Combined Features Ubuntu Answer



Method	Prec.	Recall	F1	Accu.
LM+GQL	0.726	0.718	0.722	0.695
Stopword+NG	0.735	0.786	0.760	0.726
LM+POSI	0.733	0.812	0.770	0.733
LM+Stopword	0.758	0.764	0.761	0.735
LM+AUTH	0.739	0.840	0.786	0.748
POS+Stopword	0.785	0.811	0.798	0.773
LM+POSI+Stopword	0.785	0.814	0.799	0.774
LM+POSI+AUTH	0.929	0.964	0.946	0.940
POSI+AUTH	0.935	0.969	0.952	0.946

Table 10: Combined Features DC Answer



Method	Prec.	Recall	F1	Accu.
LM+GQL	0.735	0.594	0.657	0.688
LM+AUTH	0.700	0.771	0.734	0.719
Stopword+NG	0.737	0.688	0.712	0.720
LM+Stopword	0.765	0.717	0.740	0.747
LM+POSI	0.780	0.879	0.827	0.815
LM+POSI+Stopword	0.846	0.899	0.872	0.867
POSI+Stopword	0.846	0.901	0.873	0.868
LM+POSI+AUTH	0.951	0.991	0.970	0.970
POSI+AUTH	0.958	0.993	0.975	0.975

Why does the 2nd paper cite the 1st?

- The 1st paper was the first to address the same problem
- The 2nd paper uses some of the methods proposed by the 1st one
- They compare their results with those of the 1st paper