Study of Similarity Measures as Features in Classification for Answer Sentence Selection Task in Hindi Question Answering: Language-Specific v/s Other Measures

Devika A Verma
BITS Pilani
K K Birla Goa Campus
p20160010@
goa.bits-pilani.ac.in

Ramprasad S Joshi
BITS Pilani
K K Birla Goa Campus
rsj@
goa.bits-pilani.ac.in

Shubhamkar A Joshi
Vishwakarma Institute
of Information Technology
shubhamkar.21810437@
viit.ac.in

Onkar K Susladkar
Vishwakarma Institute
of Information Technology
onkar.21810471@
viit.ac.in

Abstract

Answer sentence selection is an important sub-task in Question Answering (QA) that determines the correct answer sentence from a passage. This task can naturally be reduced to the semantic text similarity problem between question and answer candidate. In this work, we investigate the significance of various similarity measures for the answer sentence selection task in Hindi an Indo-Aryan language. *Karaka relations* is the core of dependency annotation scheme used for Hindi and are crucial to syntactico-semantic analysis of the sentence. We investigate this, and compare them to other, hitherto known measures. To investigate and compare the utility of various measures, we develop a test-bench over a benchmark Hindi and English multilingual QA corpus for comparison, making two tool-chains and designing empirical experiments across combinations of similarity measures, sentence embedding schemes, and supervised machine learning models for classification. Combining *Karaka relations* with different similarity measures shows significant performance improvement for sentence selection task, suggesting them as potentially a semantic similarity measure. Moreover, our results give us confidence that refinement of *Karaka* relations extraction to optimal quality will reduce the need for availability of large pre-trained language models.

1 Introduction

Information-retrieval-based (IR-based) QA systems typically employ a search engine and a passage retrieval module that narrows down the answer search space from a huge corpus to a small set of sentences. On the contrary, QA systems depending on reading comprehension are provided with a passage for answer extraction. In both kinds of systems, the returned answer can be a word, a phrase or a sentence. Wang et al. (2007) highlighted the importance of returning a complete sentence over a short phrase as an answer against the user query. The study conducted by Lin et al. (2003) shows that as the amount of text returned by a QA system increases, users utilize the supporting text to get the answer, significantly decreasing the number of further queries.

Answer sentence selection as an important task in QA has been a widely researched topic since the release of *QASent* and *WikiQA* corpora. *QASent* (Wang et al., 2007) was created by choosing sentences from *Text REtrieval Conference (TREC)* QA 8-13 track data(Voorhees, 1999). *WikiQA* (Yang et al., 2015), that was released later, was developed in a more natural and large setting. Researchers have sought to accomplish this task by combining (in varying degrees) techniques of natural language processing, statistical analysis and deep learning for lexical, syntactic and semantic processing of the question and candidate sentences. We can certainly say that the performance of QA systems depends largely on the quantitative and qualitative advances in computational resources specifically available for the language of QA. But then this is a key challenge for low resource languages. Languages that have less digital presence, languages in which colloquiality and dialects dominate standardized printed and digital content, languages with complicated

descriptive grammars elusive to the usual parsing schemes are such low resource languages. Many Indo-Aryan languages (mainly spoken and used in South Asia or SAARC countries) lack sizeable, curated datasets and efficient, effective computational linguistics tools. The resources available for the language-dependent processing necessary for the answer sentence selection task described above are paltry and fragmented, mostly dependent on datasets translated from English. We observe that there is need of more efficient, less-computation-intensive, less-data-dependent computational linguistics tools for these languages.

Indo-Aryan languages have relatively free word order compared to European-origin languages(Sangal and Chaitanya, 1995). They have a rich system of case-endings and post-positions (together called *vibhakti*). For the important task of answer selection in these languages, we need to choose measures that take into account their distinguishing aspects. Relatively lesser attempts have been made for QA task in Hindi as well as in other Indo-Aryan languages.

In this work, the answer sentence selection task for QA in Hindi is addressed. We explore various lexical and syntactico-semantic similarity measures between question and candidate answer sentence for accomplishing this task. We regard this as a classification problem in our work and investigate the influence of similarity measures on the answer sentence selection task in Hindi. For comparison, we perform empirical analysis over Hindi and English multilingual QA corpus, experimenting across combinations of seven similarity measures, two sentence embedding techniques and eight machine learning algorithms.

The paper is organized as follows: In the next section 2, we survey the related work for English along with development for QA in Indian languages. Section 3 defines the problem statement along with description of various similarity measures explored and evaluated. Experiment design and results on benchmark Hindi and English multi-lingual QA datasets are presented in Section 4. Conclusions and future work are discussed in the last section 5.

2 Related Work

2.1 QA in Indian languages

Since 2000, Hindi-English cross-lingual question answering systems have been demonstrated. As a part of the trans-lingual information detection, extraction and summarization program Satoshi et. al (Sekine and Grishman, 2003) developed a system that accepts an English query, searches the answer in Hindi newspapers and returns an English response. (Shukla et al., 2004) created an intermediate representation called Universal Networking Language(UNL), that makes a hyper-graph to represent the meaning of the text in source language. Thus given a source text in several languages, their system could respond without translating the source text in questioner's language.(Gupta et al., 2012) proposed a natural language interface to relational database using computational Paninian grammar and *Karaka relations* to perform semantic analysis. (Gupta et al., 2018) proposed a framework for cross-lingual factoid and descriptive QA in Hindi and English. They deployed a deep learning model for question classification and answer extraction using similarity measures like proximity score, pattern matching score, n-gram coverage score and semantic similarity score.

2.2 Answer Sentence Selection

Studies highlight that measures like overlapping n-grams, surface pattern matching or bag-of-word representations give certain level of success in answer sentence selection (Martinez et al., 2012), while they fail to determine syntactic and semantic variations between question and answer pair which should be captured for appropriate answering(Yih et al., 2013). In many QA systems syntactic relations between verb, arguments and certain type of question words are extracted from the parse tree (Shen et al., 2005). Following the approach pioneered by Punyakanok et al. (2004), question and candidate sentences were represented with their dependency tree that incorporate semantic information along with using tree-edit distance between them as selection criteria (Wang and Manning, 2010; Heilman and Smith, 2010; Shen and Klakow, 2006). Ignoring important relations between tokens in a sentence is a reason for false positives for the existing retrieval techniques (Tellex et

al., 2003) This was mainly because several irrelevant passages may share the same query keywords, but the relations between these tokens might be totally varying from relations in query keywords. (Cui et al., 2005) demonstrated that to determine the similarity between two sentences it is crucial to examine similarity between all the corresponding relation paths extracted from dependency trees. (Sultan et al., 2014)'s work highlights that dependency types may exhibit equivalence and it is necessary to develop a mapping between dependency types. Calculating best match between syntactic and semantic structured representations incurs high computational cost O(V^2L^4) (Yih et al., 2013). Later works explored named entity recognition, answer type tagging, word and phrase alignments (Sutedi et al., 2019),(Wang et al., 2007). It has been observed that performance and errors from each of the above mentioned syntax and semantic analysis modules impacts the answer selection accuracy performance. More recently, deep learning approaches tend to have overshadowed these NLP tasks that do not rely on linguistic tools, but certainly require huge corpus and computational resources(Tan et al., 2016) (Ma et al., 2015) described dependency-based convolution neural networks (CNN) which gave higher performance for question classification tasks over the baselines. Current efforts are towards learning sentence representation that captures essential information to strengthen similarity between question and candidate sentence pair (Feng et al., 2015), (Yu et al., 2014). Large scale pre-trained word vectors and multi lingual models for sentence representations have been released for some group of languages (Artetxe and Schwenk, 2019), (Sanh et al., 2020). Several baselines have been established for answer sentence selection task in English over bench mark dataset like TREC-QA, WikiQA, QASent. To the best of our knowledge this is a first attempt for the key task of answer sentence selection in Hindi QA.

3 Methodology and Implementation

3.1 Problem Definition

Given question q and a passage/context with a a set of candidate answer sentences $\{s_1, s_2 ... \ s_n\}$ for q, the problem is to identify the best matching candidate sentence s_i where $(1 \leq i \leq n)$. If the sentence identified is the ground truth answer then q is considered as correctly answered. We formulate this as a classification problem in a supervised learning setting. Each instance in the training corpus is a context and question pair, associated with a target label which indicates the index of the answer sentence within the context.Two such instances are shown in figure 1. Using this labeled dataset we learn an answer sentence selection model that predicts the target label for any new instance of context and question pair. To train this machine learning model, we transform the given candidate sentences from context to features, value of which is based on its similarity to question. The focus of this work is to evaluate the similarity variants based on model's accuracy to predict correct answer sentence from the given context. The high-level process for experimenting with different similarity variants is depicted in figure 2.

3.2 Similarity Measures

Given an instance of question and a context with *n* candidate answer sentences (candidates), we represent it using n-dimensional feature set, in the training set, the value of which is computed using similarity between the question and candidate. For each candidate, we build one feature based on its similarity to question. If a context has less than *n* sentences, we replace its feature value with a score that indicates no correlation between question and candidate, to make total *n* candidates in all contexts for uniformity. In order to compute similarity between question and candidate, we use seven different lexical and syntactico-semantic measures either individually or in combinations of each other.

3.2.1 Content Word Overlap (cwo)

The simplest measure is lexical overlap between the words of the question and candidate sentences(Gupta et al., 2018; Yih et al., 2013). For certain datasets like QASent (Wang et al., 2007), where candidate answer sentences for questions are curated by matching content words of question. Yang et al. (2015) show that simple word matching method establishes a strong baseline. However the performance may degrade for datasets arising in more natural and realistic situations. We explore this pos-

context	question	text	target
Pappataci fever is prevalent in the subtropical zone of the Eastern Hemisphere between 20Â°N and 45Â°N, particularly in Southern Europe, North Africa, the Balkans, Eastern Mediterranean, Iraq, Iran, Pakistan, Afghanistan and India.The disease is transmitted by the bites of phlebotomine sandflies of the Genus Phlebotomus, in particular, Phlebotomus papatasi, Phlebotomus perniciosus and Phlebotomus perfiliewi. The sandfly becomes infected when biting an infected human in the period between 48 hours before the onset of fever and 24 hours after the end of the fever, and **remains infected for its lifetime**. Besides this horizontal virus transmission from man to sandfly, the virus can be transmitted in insects transovarially, from an infected female sandfly to its offspring.Pappataci fever is seldom recognised in endemic populations because it is mixed with other febrile illnesses of childhood, but it is more well-known among immigrants and military personnel from non-endemic regions.	Does an infection for Sandflies go away over time?	remains infected for its lifetime	3
Parenchyma cells are living cells that have functions ranging from storage and support to photosynthesis (mesophyll cells) and phloem loading (transfer cells). Apart from the xylem and phloem in their vascular bundles, **leaves** are composed mainly of parenchyma cells. Some parenchyma cells, as in the epidermis, are specialized for light penetration and focusing or regulation of gas exchange, but others are among the least specialized cells in plant tissue, and may remain totipotent, capable of dividing to produce new populations of undifferentiated cells, throughout their lives. Parenchyma cells have thin, permeable primary walls enabling the transport of small molecules between them, and their cytoplasm is responsible for a wide range of biochemical functions such as nectar secretion, or the manufacture of secondary products that discourage herbivory. Parenchyma cells that contain many chloroplasts and are concerned primarily with photosynthesis are called chlorenchyma cells. Others, such as the majority of the parenchyma cells in potato tubers and the seed cotyledons of legumes, have a storage function.	What are made up almost entirely of parenchyma cells?	leaves	2

Figure 1: Instances from Multilingual QA Corpus (Lewis et al., 2019)

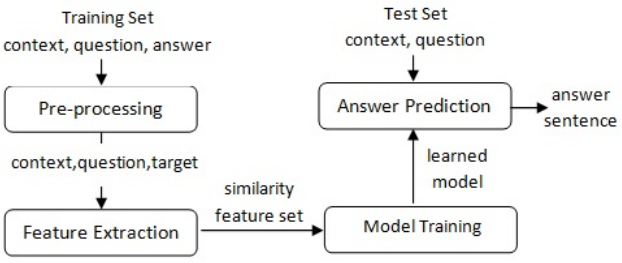

Figure 2: High Level Schematic of the Test Bench for Answer Sentence Selection

s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	target
cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	cos_l	
0.511	0.609	0.675	0.64	0.588	0.064	0	0	0	0	0	0	3
0.686	0.731	0.599	0.633	0.727	0.649	0.076	0	0	0	0	0	2

Figure 3: Cosine Similarity between Candidate Answer Sentences & Question Vectors

sibility for answer sentence selection in Hindi and English. After removing the stop-words, we count the number of words in the question that also occur in the answer sentence.

3.2.2 Longest Common Sub-string (lcs)

Longest Common Sub-string is the maximum length sub-string from all common sub-strings between two texts. In this work, we calculate the length of longest common sub-string between question and the candidate answer sentences to compute similarity.

3.2.3 Cosine Similarity (cos) and Euclidean Distance (euc)

Vector space models are widely used in IR to assess relevance of documents to queries. From simple bag-of-words models to transformer based models, there are several state of the art techniques and pre-trained models to embed the sentences into a vector (Wang et al., 2020). Multilingual LASER (Artetxe and Schwenk, 2019) and BERT BASE (Sanh et al., 2020) have large scale pre trained models that encode semantic and contextual information in sentence and have achieved state of the art performance in various NLP tasks. From our survey and initial experimentation we found these models provided a better representation for Hindi sentences over other techniques and we obtain the vector representation of question and candidate sentence using LASER and BERT BASE. To measure correspondence and similarity between these vectors following metrics are used

1. **Cosine Similarity** is a most popular metric that measures the cosine of the angle between two vectors and its value ranges between 0-1. Has been earlier used for QA similarity (Martinez et al., 2012), document clustering, plagiarism detection, IR (Han et al., 2012), (Metcalf and Casey, 2016).

2. **Euclidean distance** is another similarity measure based on the premise that every instance

s1		s2		s3		s4		s5		s6		s7		s8		s9		s10		s11		s12		target
cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	cos_l	cwo	
0.511	0	0.609	1	0.675	0	0.64	0	0.588	0	0.064	0	0	0	0	0	0	0	0	0	0	0	0	0	3
0.686	2	0.731	2	0.599	2	0.633	2	0.727	2	0.649	2	0.076	0	0	0	0	0	0	0	0	0	0	0	2

Figure 4: Cosine Similarity & Content Word Overlap between Candidate Answer Sentences & Question

can be represented as a Cartesian point in N dimensional space. Within the euclidean space, euclidean distance is the length of the line connecting two points computed using Pythagoras theorem.

Cosine similarity computed between vectors obtained using Multilingual LASER and BERT BASE models are denoted by cos_l and cos_b respectively while euc_l and euc_b denote euclidean distances. Figure 3 shows cosine similarity computed for the instances shown in figure 1. Euclidean distance is computed in similar manner.

3.2.4 Word Movers Distance (wmd)

Word movers distance is another tool in IR which enables assessing similarity between two sentences with different words(Kusner et al., 2015). It uses vector embedding of words using word2vec algorithm(Mikolov et al., 2013). We compute word movers distance between embedded words of question and candidate answer sentences , using open source SpaCy's (Honnibal and Montani, 2017) wmd implementation along with pre- trained fastText word embedding for Hindi and English words (Bojanowski et al., 2016).

3.2.5 Karaka Relations (kr)

To select correct answer sentence for the given question, understanding the meaning and context of the question and choosing the candidate sentence with similar relevant context and meaning is important. This necessitates sentence analysis. *Karaka* relations from grammar written for Sanskrit by Indian grammarian Panini in 7th century BCE, provides such a syntactico-semantic analysis of a sentence. As per Paninian framework, meaning of the sentence is encoded in the words as well the relationship between the lexical items (words) in the sentence. The main action is denoted through the verb in the sentence and the grammatical relations between words are categorized into

1. *Karaka* relations that identify direct participants in the action carried out in the sentence, example 'doer'-*karta*, 'destination/goal'-*karma*, 'instrument'-*karana*, ' recipient/beneficiary'-*sampradaan*, 'source'-*apaadaan*, 'location'-*adhikarana* of the action. (Panini identifies six *Karaka* relations).

2. Relations other than Karaka that identify words that do not have direct role in action, example -reason.

As highlighted by (Sangal and Chaitanya, 1995), Hindi language has morphological rich system of case-endings and post-positions (together called as *vibhaktis*), which act as explicit markers to identify the above participatory role a word plays in the sentence. The scheme adopted for Hindi treebanking and dependency parsing of a sentence is based on Paninian theory and the dependency annotations are categorized as *Karaka* relations and non-karaka relations. To extract these relations we obtain the dependency parse of the question using Stanza (Qi et al., 2020) and extract the following:

1. verb present at the root of parse tree to identify the main action being carried out in sentence

2. universal dependency relations (as shown in table 1) corresponding to the six *Karaka* relations to identify the direct participants in accomplishing the action

3. question word (example 'who', 'when') to identify specific *vibhaktis* (case-endings) based on set of handcrafted rules

For every candidate answer sentence within both Hindi and English contexts, we check if the action verb and direct participants are present. We check

question word from Hindi questions only as *vibhaktis* are specific to Hindi language. For English, candidate with matching root and relations while for Hindi, the candidate with matching root, relations and expected *vibhaktis* will be semantically more relevant to answer the question.

Karaka Relation	Universal Dependency
Karta / Doer	nsubj,nsubjpass, nmod,dobj
Karma / Goal	dobj, ccomp, xcomp,nmod,acl
Karna / Instrument	nmod
Sampradan / Recipient	iobj,nmod, nsubj
Apadan / Source	nmod
Adhikaran / Location	nmod

Table 1: Mapping from Karaka Relations to Universal Dependency (Tandon, 2018)

We use the similarity measures either individually or in combinations of each other to obtain the feature representation of a candidate sentence. Figure 3 show cosine similarity measures computed for three instances, while figure 4 show combination of cosine similarity and content word overlap computations. Likewise we experiment with 6 similarity measures and their combinations.

4 Experiments and Results

4.1 Experiment Design

4.1.1 Dataset

We choose Hindi from the Indo-Aryan language family for the investigation. Hindi being official language in India, is one of the widely used natural languages including in the cyberspace. Multilingual question answering (MLQA) (Lewis et al., 2019) is a multiway aligned benchmark dataset available in seven languages. MLQA is the only available sufficiently large corpus for Hindi or from Indo Aryan Language family that has 5 thousand extractive QA instances. For comparison purpose we also experiment on English MLQA corpus having over 12 thousand instances. Each instance in the corpus has context, question and answer text. Few contexts have more than one question. For sake of simplicity in experimentation, we restrict to the context length of at the most 12 sentences (around 90% of the entire corpus) so that we have 12 target labels corresponding to the sentence indexes to be predicted in this problem.

4.1.2 Model training & answer sentence prediction

Feature extracted for training the ML models are described in section 3 . For implementing all machine learning, pre-processing, training, cross-validation, we use open source python library `scikit-learn` (Pedregosa et al., 2011). Answer selection model is trained separately for Hindi and English over 23 different feature sets, using Multinomial Logistic Regression(MN), XgBoost(XgB), Random Forest(RF), K-nearest neighbours(KNN), Kernel Support Vector Machine(SVM), Decision Tree(DT), Feed Forward Neural Network(FFNN) and Convolutional Neural Network(CNN) algorithms.

4.2 Results

We performed several runs on each of the 184 combinations of our 23 feature sets and 8 ML model trained using the same pre-processed dataset, for each of the languages Hindi and English.

Comparison Yardsticks After comparing sufficiently large number of outcomes, we focused on the best results obtained on each feature set from the chosen 8 ML schemes. We found that MN, XgB and CNN are performing as the top three consistently for all feature sets. Thereafter we proceeded to compare the features sets from Hindi and English on these three ML schemes.

Comparison Outcomes and Visualisation Table 2 and 3 shows the best validation accuracy obtained over Hindi and English MLQA. We compared the results of each feature set that didn't include *Karaka* relation(kr) (second column from accuracy tables) with feature set having kr-based feature as the only additional feature combined (third column from accuracy tables). The results unambiguously establish that the performance of answer selection improves significantly when *Karaka* relation features are combined with other feature sets. Highest accuracy 64.60% for Hindi is reported when *Karaka*

Feature Set {A}	Accuracy using {A}	Accuracy using {A}+kr
cos_l	54.73%	57.73%
cos_b	45.63%	55.14%
euc_l	39.19%	54.73%
euc_b	46.33%	51.39%
wmd	41.64%	53.08%
cwo	54.63%	58.84%
lcs	45.74%	55.96%
cos_l + cwo	57.97%	60.49%
cos_b+cwo	54.72%	58.02%
cos_l+cwo+wmd	56.79%	62.96%
cos_b+cwo+wmd	56.37%	**64.60%**

Table 2: Best Answer Sentence Selection Accuracy for Hindi MLQA

Feature Set {A}	Accuracy using {A}	Accuracy using {A}+kr
cos_l	59.29%	66.18%
cos_b	53.65%	58.57%
euc_l	44.09%	61.41%
euc_b	50.54%	57.74%
wmd	59.87%	64.40%
cwo	67.01%	70.56%
lcs	48.69%	62.83%
cos_l + cwo	67.09%	69.52%
cos_b+cwo	62.20%	67.84%
cos_l+cwo+wmd	66.49%	70.77%
cos_b+cwo+wmd	67.22%	**70.98%**

Table 3: Best Answer Sentence Selection Accuracy over English MLQA

relation features are combined with cosine similarity, word movers distance and content word overlap (cos_b+cwo+wmd+kr). Moreover, *Karaka* relation as a feature set, alone reported accuracy of 57.22% which is better than every other single-similarity feature set.

For English, highest accuracy 70.98% is obtained when *Karaka* features are combined with content word overlap (cwo+kr). Further adding cosine similarity and word movers distance features (cos_b+wmd+cwo+kr) did not show significant improvement in performance over cwo+kr. Adding euclidean distance and longest common sub-string somewhat degraded the overall performance for both English and Hindi. All these results are presented in the graphs in Figure 5. There are 11 feature sets (without *Karaka* relations), and the change in accuracy over *Karaka* relations when augmented by each of them is in the red (for Hindi) and yellow (for English) bars. Whereas the improvement in accuracy for each of those sets when *Karaka* relations augments them is in the adjoining 11 bars (blue–Hindi, green–English).

4.3 Interpretation and Discussion

Our outcomes suggest that *Karaka* relations can be a significant similarity measure for Hindi.

- ***Karaka* relations Alone or Combined Wins** Figure 5 is visual representation of comparison of the contribution to accuracy improvement of features when used in combinations. The bars show the %age change in accuracy when an additional feature augments a feature set. Here features are the 6 individual features (similarity measures), among which cosine and euclidean measures are further split into two each according to the underlying vectorisation (LASER or BERT). The bar heights h are calculated thus: if $A = \{s_1, s_2, \dots\}$ is a feature set, and we want to see the contribution of the subset $B \subsetneq A$, then $h_{B/A} = 100(f(A) - f(A \backslash B))/f(A \backslash B)$ where $f(S)$ is the accuracy of the feature set S.
  It is evident that *Karaka* relations enhances accuracies significantly across the board when it augments any feature set, whereas the improvement in accuracy when other feature sets augment *Karaka* relations (alone) is less or negative.
- **Feature Overkill Does Not Help** We can see that for single features the improvement that can be attributed to *Karaka* relations, when augmented, is significant to large. Feature sets without *Karaka* relations do not improve accuracy with addition of more features. For English, euclidean distance measure seems to contribute much less to accuracy when combined with others, and *Karaka* relations completely dominates it. In the other direction, adding *Karaka* relations to a large feature set enhances

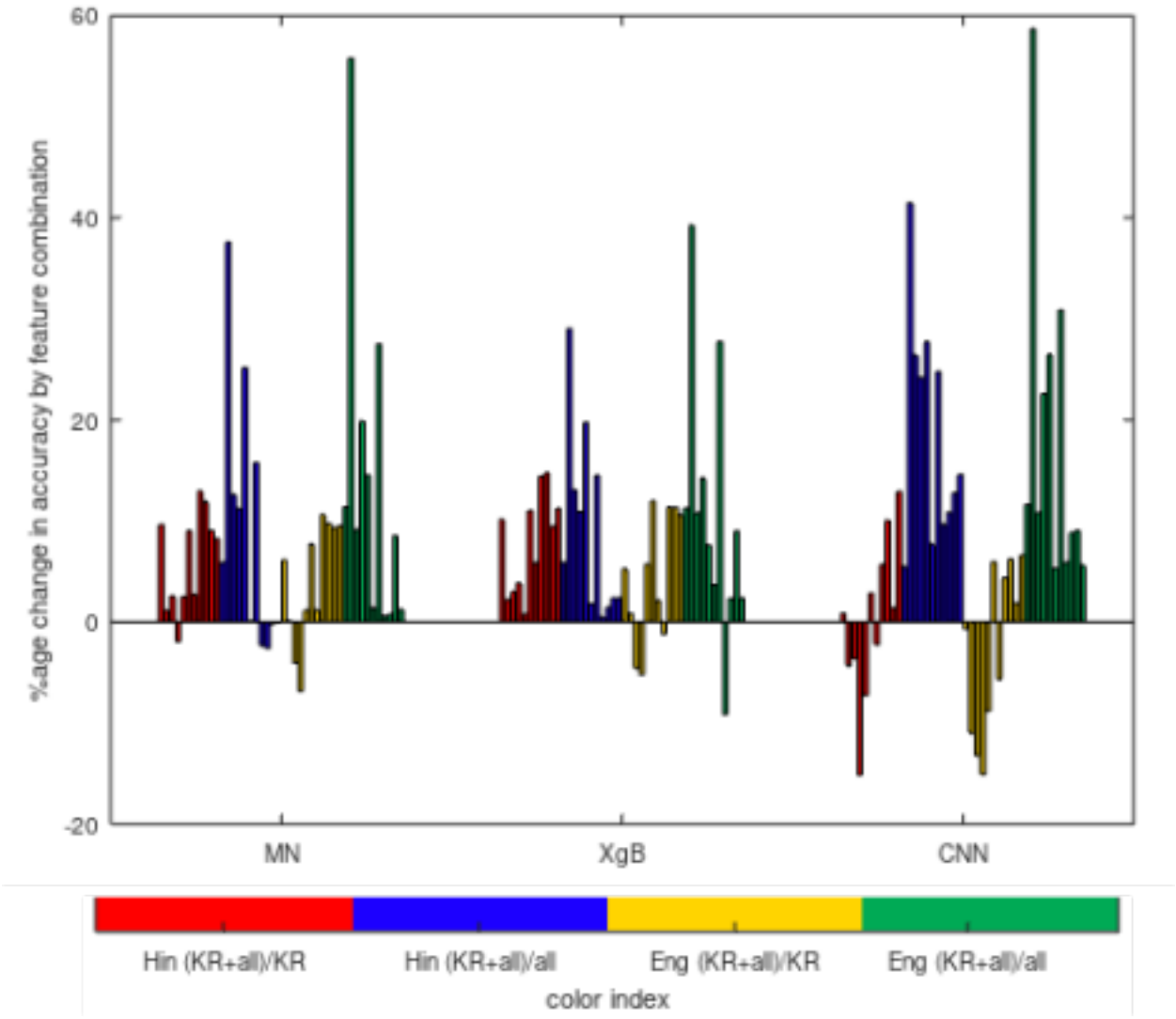

Figure 5: Accuracy Enhancement or Reduction by Combinations

The feature sets (with and without kr), left-to-right, for each 11-bar group of the same color: 1. cos_l, 2. euc_l, 3. cos_b, 4. euc_b, 5. wmd, 6.cwo, 7. lcs, 8. cos_l+cwo, 9. cos_l+cwo+wmd, 10. cos_b+cwo, 11. cos_b+cwo+wmd

the performance greatly, that gain has not been seen while the feature set grew without *Karaka* relations.

- **Semantic Similarity** The similarity measures that do contribute to accuracy enhancement when combined with *Karaka* relations are expected to capture semantic information indirectly. Karaka relations, when exploited fully, will help capture semantic and context information directly. In our experiments we have not yet attempted that, we have not exploited the full potential of *Karaka* relations. Therefore, these results suggest that research in this direction – exploiting *Karaka* relations fully to capture semantic information directly – can possibly reduce computational overheads in feature-heavy models by eliminating other features.

5 Conclusion

In this work, we investigate the influence of various similarity based feature sets on answer sentence selection task in Hindi QA, and outcomes highlight that *Karaka* relations can be a significant similarity measure. We have not exploited the full potential of using *Karaka* relations to obtain the syntactico-semantic sentence analysis for measuring question and answer candidate similarity. Doing so is not computationally intensive, but the morphological analysis, disambiguation, lexical analysis, and different parsing actions that are needed to go the whole hog requires extensive linguistic knowledge converted into computational models. If this is sought to be done using ML again, then choosing or curating relevant datasets with sufficient variety of examples including ambiguity and other linguistic challenges will need, in turn, the same extended linguistics expertise. Our experimentation and visualisation of the outcomes surely makes a strong case to attract this investment into this enterprise. Precisely, we demonstrate that such linguistic knowledge, incorporated into *Karaka* relations extraction and use of the same for similarity measures in the way we did, can reduce computational costs for the same accuracy goals and the similar datasets. Moreover, refinement of *Karaka* relations extraction to optimal quality will definitely reduce the need for availability of large pre-trained language models.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. volume 39, pages 400–407, 08.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task.

A. Gupta, A. Akula, D. Malladi, P. Kukkadapu, V. Ainavolu, and R. Sangal. 2012. A novel approach towards building a portable nlidb system using the computational paninian grammar framework. In *2012 International Conference on Asian Language Processing*, pages 93–96.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A multi-domain multi-lingual question-answering framework for English and Hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Jiawei Han, Micheline Kamber, and Jian Pei. 2012. Getting to know your data. In *Data Mining*, pages 39–82. Elsevier.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019, Los Angeles, California, June. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 957–966.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. What makes a good answer? the role of context in question answering. In *Proceedings of INTERACT 2003*, pages 25–32.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based convolutional neural networks for sentence embedding.

David Martinez, Andrew MacKinlay, Diego Molla-Aliod, Lawrence Cavedon, and Karin Verspoor. 2012. Simple similarity-based question answering strategies for biomedical text. *CEUR Workshop Proceedings*, 1178:1–13.

Leigh Metcalf and William Casey. 2016. Metrics, similarity, and sets. In *Cybersecurity and Applied Mathematics*, pages 3–22. Elsevier.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2004. Mapping dependencies trees: An application to question answering. In *In Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics, Fort*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Rajeev Sangal and Vineet Chaitanya. 1995. *Natural Language Procssing: A Paninian Perspective*. Prentice Hall of India.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Satoshi Sekine and Ralph Grishman. 2003. Hindi-english cross-lingual question-answering system. *ACM Transactions on Asian Language Information Processing*, 2(3):181–192.

Dan Shen and Dietrich Klakow. 2006. Exploring correlation of dependency relation paths for answer extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 889–896, Sydney, Australia, July. Association for Computational Linguistics.

Dan Shen, Geert jan M. Kruijff, and Dietrich Klakow. 2005. Exploring syntactic relation patterns for question answering. In *In Proc. Of IJCNL'05*.

Pushpraj Shukla, Amitabha Mukherjee, and Achla Raina. 2004. Towards a language independent encoding of documents: A novel approach to multilingual question answering. *In Proceedings of the 1st International Workshop on Natural Language Understanding and Cognitive Science, NLUCS*, pages 116–125.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230.

Ade Sutedi, Moch. Arif Bijaksana, and Ade Romadhony. 2019. Answer selection using word alignment based on part of speech tagging in community question answering. *Journal of Physics: Conference Series*, 1192:012035, mar.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Lstm-based deep learning models for non-factoid answer selection.

Juhi Tandon. 2018. Advancements in dependency parsing for indian languages. Master's thesis.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. Association for Computing Machinery.

Ellen M. Voorhees. 1999. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82.

Mengqiu Wang and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1164—-1172. Association for Computational Linguistics, August.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Lili Wang, Chongyang Gao, Jason Wei, Weicheng Ma, Ruibo Liu, and Soroush Vosoughi. 2020. An empirical survey of unsupervised text representation methods on Twitter data. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 209–214, Online, November. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, September. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753, Sofia, Bulgaria, August. Association for Computational Linguistics.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection.