Team Contribution

The group of three worked in parallel to the final form of the report and the poster. Towards the end of the project cycle the team contributions could be subdivided into :

Kaili Wang : Kaili contributed to the code development and testing framework. Kaili led the group in iterable changes to the code : such as updating the number of epochs, and specific POS to replace.

Yara : Yara contributed to the background and future research. Yara led the group in researching the scholarship of data augmentation using POS replacement. She structured the conclusion of the research as well as the future.

Alexis : Alexis contributed methodology and approach formation. Alexis led the group to the approach of leveraging pos tagging and synonym substitution via spaCY and wordNet NLTK; she set up a framework to incorporate these libraries/methods into the baseline code

Question Answering Augmentation System: Conditional Synonym Replacement

Stanford CS224N Default Project

Alexis Mack
Department of Computer Science
Stanford University
asmack@stanford.edu

Yara Sevilla
Department of Computer Science
Stanford University
yarasev6@stanford.edu

Kaili Wang
Department of Computer Science
Stanford University
kkwang@stanford.edu

Abstract

In this project, our team implemented Data Augmentation tools, such that we can distinguish and target different parts of speech and bolster the data with synonym replacements. We also explore whether augmenting just the context or augmenting the context, question, and answer provides better EM and F1 results. We hope to speed up data augmentation and experiment with the number of epochs for testing and the targeted words to augment.

1 Key Information to include

- Mentor: Kendrick Shen
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

For human-computer interaction, natural language is the best information mechanism for humans. Natural language system present issues in answering specific questions of users. The limitation inspired Question Answering Systems. Question Answer Systems aim to automatically understand and answer a question originally in a natural language context (Aroussi et al., 2016). Question Answer systems (QAS) have an advantage over search engines in meeting a user's information needs. Still the quality of Question Answering (QA) is still not sufficient regarding the number of questions that are answered correctly. The research problem is how the QA quality can be improved.

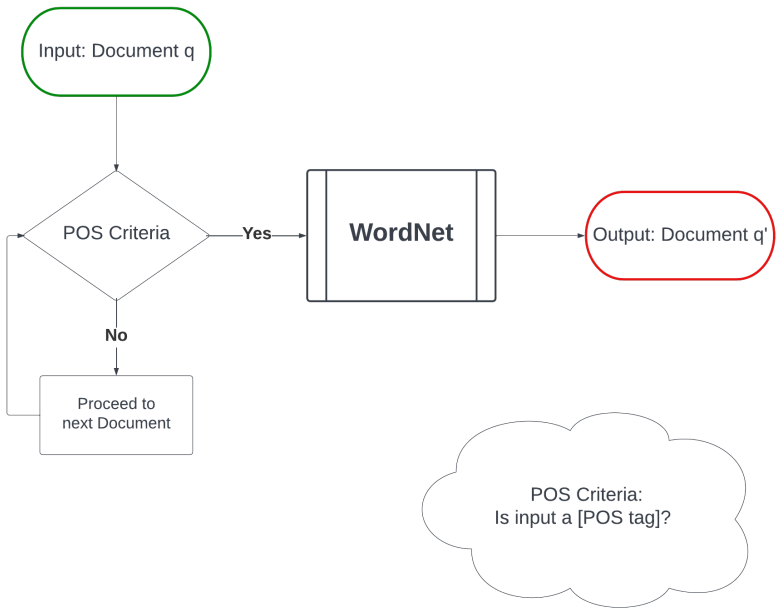
Currently most QAS approaches are applied to several domains. In most question answering systems, different systems of inputs are relevant for different type of questions. The difficulty here is that Domain specific QA systems often do not have enough training examples to train a robust and accurate model.

This means that large amounts of varying pre-labelled and pre-processed data are necessary to train and test the robustness of these QA models. However, this relevant data could be difficult, expensive, or impractical to collect. Data augmentation is one avenue to extrapolate more meaning from few data-points. Examples of data augmentation improving machine learning problems include image

classification, computer vision, and speech (Krizhevsky et al., 2017; Cui et al., 2015).

Also, adding more training examples via manual labor is expensive and time consuming. However, creating new datasets with already found, labelled data with algorithms are relatively inexpensive. Therefore, our approach will be to create more input data-points out of the already given in-domain and out of domain datasets using spaCy, an open-source software library for advanced natural language processing and WordNet, a lexical database of semantic relations between words in more than 200 languages (Honnibal, 2015; Wei et Zou, 2019).

Many current works examine synonyms as a way to augment data, yet they have not explored the importance of parts of speech (POS) in determining closeness of similarity scores (Murata et al., 2019; M. Wang et al., 2016). Based on these works, we examine the differences in augmented data with different POS Criteria. Below is a figure highlighting the workflow in how we sequentially augment data:



By testing different augmented datasets and examining their EM and F1 scores, we improved the baseline model in the given BERT QAS. Synonym replacement has proven to be a reliable data augmentation path, along with context based synonym insertion-before answer, context based synonym insertion-after answer, context chunk swapping, and context deletion (Salman, 2020). However, adding the dimension of parts of speech mean that we can control the data more to prevent more noise from entering our inputs.

3 Related Work

In our research, we found many works where data augmentation bolsters the given training datasets to make stronger correlations within the neural network. Data augmentation can also be used to make the training set more domain agnostic, allowing for better out of domain testing results (Longpre et al., 2019). There are several methods to augment collected data. One method to increase the number of datapoints includes back-translation. For example, given a passage q, back-translation would create a new q_0, by translating q to a pivot language, such as Russian, and then back to a new English paraphrase (Ng et al., 2020). Other methods include random insertion, random swap, and random deletion, which can all be classified as word substitution (Wei et Zhou, 2019).

Word substitution can be implemented in different ways. One way includes consulting a lexicon that has previously mapped word classifications, such as WordNet or Word2Vec (Wei et Zhou, 2019). Another creates word with `[MASK]` tokens, which are then replaced by a pre-trained algorithm, such as BERT, with appropriate new words (Garg et Ramakirshnan, 2020). One challenge to these methods is that, because they are machine developed data, they may have incorrect sentiments or add unnecessary noise to the performance of the algorithm. Nevertheless, previous papers point to data augmentation as a viable path in training more robust question answering models (Salman, 2020; Yi, 2020).

4 Approach

Often neural models learn brittle correlations that hurt their outof-domain performance. We are augmenting the data in our approach to avoid brittle correlations. The follow depict the steps of the single data augmentation process:

Given an input x = (context, question) we create and add an input x' = (context', question') to the data set. Here the data set is augmented with additional data rather than data replacement.

Data augmentation via substitution is one of the ways to deal with labeled data scarcity and over fitting. When training neural networks for classification we rely on synonym replacement as a method of data augmentation. The augmentation is based on the substitution of words using the Word-Net corpus.

Rather than replacing generic words with synonyms we focus on replacing specified parts of speech with their synonym. Part of speech (POS) tagging is the process of categorization of word in a corpus corresponding to a part of speech.

POS tagging applies syntactic category to each word in the context question and answer. English has 9 main categories : verb, noun, pronoun, determiner, adjective, adverb, preposition, conjunction and interjection. We rely on the spaCY library to implement POS tagging. spaCY returns a Doc object from raw text that comes with annotations. spaCY offers annotations of universal POS tag. For each token in the raw text. `Token.pos_` returns the part of speech.

The results are as follows : PRON, NOUN, AUX, VERB, ADP, DET, NOUN, ADJ

For all tokens with `Token.pos_` equal to ADJ NOUN or VERB we carry out synonym replacement

Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Synsets are interlinked by means of conceptual-semantic and lexical relations. The words in a synset are lemmas

Note : We do not need WordNet to find proper nouns : use the Part-Of-Speech tagger `pos_tag`

In order to find synonyms for the Lemmas we rely on NLTK synset functions to retrieve synonyms. Given synonyms for each lemma we retrieve the synonym with the highest similarity score.

Our baseline replaces instances of VERB and ADJ with synonym. We add this replacement to our to dataset, along with the original correlated question/answer. For every context, we create only one augmented context (by converting each verb and adjective to its synonym). Thus, our dataset doubles in size. One note is that even if the converted word is in the question or answer, we leave the question/answer untouched. For example:

Context: "The dog was wet"
Question: "Who was wet?"
Answer: "The dog"

Augmented context: "The dog was damp"
Question: "Who was wet?"
Answer: "The dog"

After learning from our baseline, we replaced instances of VERB and ADJ with synonyms in the context, question and answer. We add this replacement to our to dataset, along with the original correlated question/answer. For every QA pair, we create an augmented question, answer and context (by converting each verb and adjective to its synonym). Still, our dataset doubles in size. For example:

Context: "The dog was wet"
Question: "Who was wet?"
Answer: "The dog"

Augmented context: "The dog appeared damp"
Question: "Who appeared damp?"
Answer: "The dog"

Once more we applied learning to add more complexity to the baseline, we replaced instances of NOUN, VERB and ADJ with synonyms in the context, question and answer. We add this replacement to our to dataset, along with the original correlated question/answer. For every QA pair, we create an augmented question, answer and context (by converting each nounm verb and adjective to its synonym). Again, our dataset doubles in size. For example:

Context: "The dog was damp"
Question: "Who was wet?"
Answer: "The dog"

Augmented context: "The canine appeared damp"
Question: "Who appeared damp?"
Answer: "The canine"

We train with the augmented dataset. Our performance is improved by creating additional correlations.

5 Experiments

5.1 Data

We use the given dataset for the RobustQA default project. We implement a synonym replacement algorithm that targets different POS. Also, we distinguish datasets that augment just the contexts given, or whether the context, question, and answer (CQA) are all augmented. These are listed below:

- **Augmented dataset A:** targets verbs (all words with VERB label) and adjectives (all words with ADJ label) and replaces only the contexts with the highest similarilty score in all training datapoints, bolstering the training data by 2X
- **Augmented dataset B:** targets verbs (all words with VERB label) and adjectives (all words with ADJ label) and replaces the CQA with the highest similarilty score in all training datapoints, bolstering the training data by 2X
- **Augmented dataset C:** targets nouns (all words with NOUN label), targets verbs (all words with VERB label) and adjectives (all words with ADJ label) and replaces the CQA with the highest similarilty score in all training datapoints, bolstering the training data by 2X

Below are the given original dataset values to understand the difference in in-domain dataset values and the out of domain dataset values.

Dataset	Question Source	Passage Source	Train	dev	Test
		`in domain dataset`			
SQuAD	Crowdsourced	Wikipedia	50000	10507	-
NewsQA	Crowdsourced	News articles	50000	4212	-
Natural Questions	Search logs	Wikipedia	50000	12836	-
		`out of domain dataset`			
DuoRC	Crowdsourced	Movie reviews	127	126	1248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	127	2693

Table 1: Dataset used for the project

5.2 Evaluation method

We are using two evaluation scores to determine the robustness of the Question Answering System (QAS):

- **Exact Match (EM):** A Boolean representation for each datapoint such that the given answer matches the expected answer verbatim. For example, if our expected answer is "Dog", but the given answer is "Dogs", the EM for this datapoint would be `False`.
- **F1:** A harmonic representation of precision and recall, such that there is some allowance for given answers that are close to the expected answer. For example, if our expected answer is "George Washington", but the given answer is "Washington", we would get an F1 score of 2 x prediction x recall / (precision + recall) = 2 x 50 x 100 / (100 + 50) = 66.67%.

5.3 Experimental details

5.4 Results

Based on the different Augmented Datasets, we ran `BERTForQuestionAnswering` using Microsoft Azure. The different datasets took a couple of hours to run each, but for Datasets B and C, we created a milestone on previously run data so that those took around an hour to run each. We changed the amount of epochs for Dataset C as well, to see if this caused any effect on the EM and F1 scores, after we received very similar EM and F1 scores for them. Below are these scores found:

Augment Strategy	**EM Score**	**F1 Score**
Baseline	47.51	30.63
Augmented Dataset A	**49.88**	**34.55**
Augmented Dataset B	48.77	32.20
Augmented Dataset C	48.77	32.20
Dataset C (30 epochs)	14.92	34.13

Table 1: EM and F1 Scores of the augmented datasets, defined above.

Our original hypothesis was that the datasets with augmented context, question, and answer (CQA) would perform better than the datasets with augmented contexts only. However, as shown by the table above, we see the best results with only context along with augmented adjectives and nouns. We were surprised to see the lowest match with the increased epochs, as we were expecting these more frequent changes would make the evaluation better. We explore further possible reasons for

these unexpected results in the following section.

6 Analysis

6.1 Inaccurate Augmentation Consequences

We observed some interesting outcomes of our augmentation strategy with synonym replacement, and thus wonder what the downstream consequences are. For example, this was one synonym autoplacement (only showing the first 200 characters):

Original Data:"With the example of the Ming court's relationship with the fifth Karmapa and other Tibetan leaders, Norbu states that Chinese Communist historians have failed to realize the significance of the religi..."

Dataset A:
"With the example of the Ming court's relationship with the fifth Karmapa and other Tibetan leaders, Norbu states that Taiwanese Communist historians have failed to realize the significance of the reli..."

It changed "Chinese" to "Taiwanese." Obviously, in context, this not an accurate replacement, and furthermore there are political implications and assumptions that this is more due to the fault of the NLTK synonym database, but we should be aware of such flaws of automated replacement.

6.2 Indistinguishable Augmentation Changes Between Datasets B and C

We found that throughout the augmented datasets, there was not a significant increase in EM or F1 scores in the change among the different parts of speech (POS), especially when we changed all contexts, questions, and answers (CQA) instead of just the contexts. For example, these are two different datapoints from the augmented Datasets B and C, highlighting their similarities, despite having different POS criteria:

Dataset B:
Question: "Forbes.com placed Notre Dame at what position compared to other US research universities?"
Answer: "8th"

Dataset C:
Question: "forbes.com placed notre dame at what position compared to other us research university?"
Answer: "eighth"

Because of these minimal differences, we found that the EM and F1 scores do not vary between the two Datasets B and C.

6.3 Indistinguishable Augmentation Changes Between Datasets A and B

Additionally, we found that changing just the context had counter-intuitive results; we expected this to be worse than changing the CQA, but it had better EM and F1 scores, as shown in the table. After examining the dataset, we found that these correlations between synonyms made the evaluations on Dataset A perform better. To analyze this, we looked at how many answers were changed in Dataset B to explain why Dataset B performed worse. We found that only 215 total datapoints (in domain data and out of domain data) have augmented answers; the majority of them remained the same. We hypothesized that these minimal differences formed more brittle connections that prevented the dataset from being domain agnostic.

7 Conclusion

We observed that there seems to be some small improvement in the EM score and the F1 score. Our baseline EM score is 47.51 and our baseline F1 score is 30.63. Our best augmented EM score is 49.88 and our best augmented F1 score is 34.55. These are evaluated from our Dataset A, augmented contexts of verbs and adjectives. Our baseline train NLL score is 0.9074 and our augmented train NLL score is 0.9161.

One primary limitation of our work was the time it took to train our datasets. We severely limited the possibility of augmentations on our datasets in order to conserve time for the most amount of changes with relation to the part of speech (POS) tags and the change in context and CQA.

For future experiments, we hypothesize that there are many ways to improve using these augmentation techniques. Some easy possible examples are increasing the number of synonyms given by WordNet, applying different weights to new data, depending on how many different words there are to the original dataset, and utilizing different POS tagging methods to distinguish gerunds or proper adjectives.

Another possibility is exploring how these new Supervised Learning examples could be applied to the Test Time Training algorithm. Creating new question-answer pairs could be a viable auxiliary task to the larger task of question answering, such as rotation to image classification (Sun et al., 2020).

In other words, there are larger implications to Data Augmentation, and specifically synonym replacement, to the Question Answering problem.

References

[1] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. *CoRR*, abs/1909.13231, 2019.

[2] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. *CoRR*, abs/1912.02145, 2019.

[3] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[4] Masaki Murata, Yoshiki Goda, and Masato Tokuhisa. Automatic selection and analysis of synonyms in japanese sentences using machine learning. 2014.

[5] Sina J. Semnani, Kaushik Ram Sadagopan, and Fatma Tlili. Bert-a: Fine-tuning bert with adapters and data augmentation. 2020.

[6] Li Yi. Avengers: Achieving superhuman performance for question answering on squad 2.0 using multiple data augmentations, randomized mini-batch training and architecture ensembling. 2020.

[7] Mohammed Salman. Gaining more from less data in out-of-domain question answering models. 2020.

[8] Abdelghani Bouziane, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. Question answering systems: Survey and trends. *Procedia Computer Science*, 73:366–375, 2015. International Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015).

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

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

[11] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy, July 2019. Association for Computational Linguistics.

[12] Samrat Dutta, Shreyansh Jain, Ayush Maheshwari, Ganesh Ramakrishnan, and Preethi Jyothi. Error correction in ASR using sequence-to-sequence models. *CoRR*, abs/2202.01157, 2022.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]