Generation of English Question Answer Exercises from Texts using Transformers based Models

1st Gonzalo Berger
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
gonzalo.berger@fing.edu.uy

2nd Tatiana Rischewski
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
tatiana.rischewski@fing.edu.uy

3rd Luis Chiruzzo
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
luischir@fing.edu.uy

4th Aiala Rosá
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
aialar@fing.edu.uy

***Abstract*—This paper studies the use of NLP techniques, in particular, neural language models, for the generation of question/answer exercises from English texts. The experiments aim to generate beginner-level exercises from simple texts, to be used in teaching ESL (English as a Second Language) to children. The approach we present in this paper is based on four stages: a pre-processing stage that, among other basic tasks, applies a co-reference resolution tool; an answer candidate selection stage, which is based on semantic role labeling; a question generation stage, which takes as input the text with the resolved co-references and returns a set of questions for each answer candidate using a language model based on the Transformers architecture; and a post-processing stage that adjusts the format of the generated questions. The question generation model was evaluated on a benchmark obtaining similar results to those of previous works, and the complete pipeline was evaluated on a corpus specifically created for this task, achieving good results.**

***Index Terms*—NLP for language teaching, question & answering, transformers, neural language models**

I. Introduction

The design of computer-based educational activities requires certain computer skills, as well as considerable time for their creation and revision. Teachers of different disciplines, particularly language, can benefit from the existence of tools that automate much of this process. Natural Language Processing (NLP) can be very useful in these tasks, since it allows automatic analysis of language, covering different levels: phonetics/phonology, morphology, syntax, semantics and, to a certain extent, pragmatics. The techniques and resources of this area are particularly useful for processing written texts, so they have great potential for generating educational activities based on texts selected by teachers.

The contributions of NLP to teaching can be very varied [1]: it allows the generation of educational activities, such as didactic games or classic exercises, it also helps in the automatic correction of students' work, it can be applied for analyzing discussion forums, among other applications.

In this paper we focus on the application of NLP for the generation of a particular type of activity: question and answer exercises. We study the use of NLP techniques, in particular, neural language models, for the generation of question/answer pairs from English texts. The experiments seek to generate beginner level exercises from simple texts, motivated by the need to support the universalization of ESL (English as Second Language) teaching to children in Uruguay. The work starts from a previous approach, based on the generation of questions from manually created templates [2], and seeks to explore other techniques that are currently state of the art in NLP tasks. Our approach, based on neural language models, is complemented by a preprocessing stage, which includes co-reference resolution and semantic role labeling. The complete pipeline was evaluated on simple English texts, showing very encouraging results.

The rest of the paper is structured as follows: section II describes the related work; section III gives an overview of the approach, which is structured in four stages: pre-processing, answer selection, questions generation, and post-processing; section IV describes the experiments carried out for the question generation stage, section V describes the evaluation of the approach; and, finally, section VI shows some conclusions and future work.

II. Related Work

In the area of Automatic Question Generation (AQG), the traditional approach has been the definition of templates and rules, to be applied on sentences or texts pre-processed with linguistic tools [3]–[5]. Most of the work has focused on generating wh-questions from simple sentences, aiming at the evaluation of text comprehension. Some authors have researched the generation of questions from sentences with more complex structures [6] and questions aiming at the assessment of grammatical concepts [7].

In recent years, the availability of datasets for training machine learning models, mainly the Stanford Question Answering Dataset (SQuAD) corpus [8], [9], has allowed the experimentation with neural networks. Encoder-decoder with attention architectures have been frequently used for AQG. Du et al. [10] was one of the first works using this scheme, they evaluate two strategies: taking sentence-level information and taking paragraph-level information. They published a partition of the SQuAD corpus that has been used as a benchmark in later works. Zhou et al. [11] also use an attention-based encoder-decoder model, they take as input the sentence, answer position information, and linguistic information (POS and

NER). They also use a mechanism to directly copy rare words from the original sentence. Du and Cardie [12] attach to each input pronoun the most "representative" antecedent given by a co-reference resolution module. Song et al. [13] detect which words in the text are relevant in the context of the question. Liu et al. [14] rely on the prediction of potential words to appear in the target question, in addition to other information such as lexical features and answer position indicators. Dong et al. [15] present the Unified Pre-trained Language Model (UNILM), which is a transformer-based pre-trained language model that can be finetuned for question generation tasks. They take as input the text and span of the answer, and generate as output a question for that answer.

In a previous work [2], we created a system for generating question-answer pairs from an English text using a rule-based approach. The text was enriched with linguistic information, using tools for POS (Part of Speech) tagging, Named Entity Recognition (NER), Semantic Role Labeling (SRL), co-reference resolution and WordNet [16] hypernyms. This information was used by a set of rules that matched different patterns in the text to create *what*, *where*, *who*, *when* and *what color* questions. We also created a small set of simple texts with manually curated questions and answers to test the automatic system, this set was extended in this work to make better evaluations.

In the current work, as we will see, we approached the problem from a machine learning perspective, experimenting with transformer-based models. The transformer [17] is a neural network architecture that uses several self-attention layers to produce a representation of the text input that can be used by other tasks. Since their introduction, transformers have become the state of the art in many NLP tasks, and the currently used models such as BERT [18] and GPT [19], and T5 [20] are based on this architecture. The T5 (Text-to-Text Transfer Transformer) model is a unified framework that treats different text-based language problems as a text-to-text problem, in contrast to BERT type models that only take as output a class label or a span of the input text.

III. Pipeline for generating question/answer pairs

To solve the problem of generating question/answer pairs from a text, we created a pipeline with four stages, where each stage uses different tools to generate an output that is the input of the next stage. Figure 1 shows a diagram of this process.

The first stage is a pre-processing of the input text, including co-reference resolution from AllenNLP [21]. Co-reference resolution makes it possible to find the antecedent of anaphoric elements, such as pronouns, which are meaningless without their antecedent. For example, in the text "Adriana is tall. She has brown hair", the pronoun "she" refers to "Adriana". The rationale behind this stage is that, without the step of co-reference resolution, the models could extract answers or generate questions that include pronouns or other words that refer to elements that are not present in the question-answer pair, making the pair not self-contained. In section V we will see that this is indeed the case.

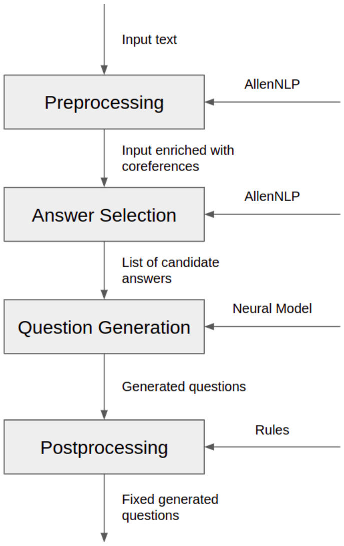

Fig. 1. Diagram of the question/answer pair generation pipeline.

The second stage selects candidate fragments from the text that could be used as answers. We use a Semantic Role Labeling module, also from AllenNLP, to select certain semantic roles as candidate answers. Semantic roles provide information about the arguments of verbs, such as Agent (element that performs an action), Patient (element affected by the action), Theme (topic being predicated about), etc. In this work, we use the standard PropBank nomenclature [22] for the identification of semantic roles: ARG0 (Agent), ARG1 (Patient), etc. We tried different combinations of roles from ARG0 to ARG4, and ARGM-TMP and ARGM-LOC, to extract answers, and compared the different results to get an optimal set of roles.

The third stage takes the text and a candidate answer, and generates possible questions from it using transformer-based models, in our case we use a model based on the T5 architecture [20]. Two corpora are available for this task, they described in detail in section IV.

The fourth stage performs post-processing to correct errors generated in the previous stage. We observed that the question generator models produce some kinds of errors that could be easily solved, such as creating questions with many repeated question marks, concatenated questions, repeated questions for the same answer.

Consider the following text:

Adriana is tall. She has brown hair. Mathew is eleven. He likes basketball.

Below is an example of the execution of the pipeline for this text, showing the output of each one of the stages:

- After the pre-processing stage, a text is obtained with its co-references solved: *Adriana is tall. Adriana has brown hair. Mathew is eleven. Mathew likes basketball.*
- In the question selection stage, a number of candidate answers are obtained from the text: *['Adriana', 'tall',*

'Adriana', 'brown hair', 'Mathew', 'eleven', 'Mathew', 'basketball']. It is worth mentioning that 'Adriana' and 'Mathew' appear twice because they are two distinct instances within the same text.

- The question generation stage generates one or more questions for each of the candidate answers obtained from the previous part. The maximum number of questions to generate for each answer is a parameter of the model: *['question': 'Who is tall?', 'answer': 'Adriana', 'question': 'Who likes basketball?', 'answer': 'Mathew',. . . other questions, . . .]*
- Finally, in the post-processing stage, possible malformations generated in the questions are fixed: we remove extra question marks, text that appears after the question mark (in some cases more questions), duplicate questions. Before post-processing: Q1) *Who likes basketball??????* Q2) *Who likes basketball? Who is tall? Who has brown hair?* After post-processing: Q1) *Who likes basketball?*

IV. Question Generation using Transformers

The question generation stage is applied after the pre-processing stage and the answer selection stage. Therefore, in this stage the input is the text with the co-references solved and the answer candidates annotated. As output, one or more questions (defined as a parameter) are generated for each answer candidate received.

In a first experimental stage we evaluated different transformer-based models and based on primary results we chose the T5 model for further experimentation. We discarded other options at an early stage since we did not have the processing capacity to perform exhaustive tests with different models, and the initial experiments with other architectures yielded poor results.

We worked with a T5-based model pre-trained on SQuAD for the question generation task [20]. This model is based on the small variant of T5, with 60 million parameters, and is available at HuggingFace as t5-small-gq-hl[1]. From now on we call this model "T5 Base".

The "T5 Base" model, already pre-trained with the SQuAD corpus, was fine-tuned in two different ways: On the one hand, only with the NewsQA corpus ("T5 NewsQA" model) and on the other hand, with a combination of SQuAD and NewsQA ("T5 SQuAD+NewsQA" model).

To train the models we used the PyTorch library [23], the hyperparameters used for training are the following: Learning rate: 10-4; Batch size: 16; Number of epochs: 3; Steps of gradient accumulation: 8; Seed: 42.

Table I shows an evaluation of the question generation stage in isolation on the SQuAD partition used as a benchmark by several works mentioned in section II. We show the BLEU [24], METEOR [25] and ROUGE$_L$ [26] metrics, widely used to assess quality in text generation tasks, which measure the similarity between a set of expected texts and a set of candidates generated by a system. BLEU calculates the similarity based on token n-grams, METEOR also includes information on the alignment and considers paraphrases, and ROUGE$_L$ is based on the longest common subsequence of tokens. In the next section we describe an evaluation of the whole pipeline, performed on a corpus of texts appropriate for teaching English as a second language at the beginning level.

Model	BLEU	METEOR	ROUGE$_L$
T5 Base	14.6	23.1	36.0
T5 NewsQA	13.5	20.8	35.0
T5 SQuAD+NewsQA	16.3	24.6	38.4
Du et al. [10]	12.3	16.6	39.8
Du and Cardie [12]	15.2	19.1	-
Song et al. [13]	14.0	18.8	42.7
Liu et al. [14]	17.6	21.2	44.5
Dong et al. [15]	22.1	25.1	51.1

TABLE I
Comparison of the different systems over SQuAD.

As can be seen from the table, the model T5 SQuAD+NewsQA outperforms T5 Base and T5 NewsQA. Moreover, in some metrics it outperforms several previous models, except for Dong et al. model [15], which obtains better results.

Table II shows the results of our three models on the NewsQA partition for test.

Model	BLEU	METEOR	ROUGE$_L$
T5 Base	4.7	17.9	23.5
T5 NewsQA	9.3	19.6	30.7
T5 SQuAD+NewsQA	8.4	17.9	29.0

TABLE II
Comparison of the trained models over NewsQA.

Tables I and II show different performances of our three models when evaluated on the SQuAD and NewsQA test corpora respectively. While the T5 NewsQA+SQuAD model performs the best on the SQuAD corpus, the T5 NewsQA outperforms the other models on the NewsQA test corpus. In addition, in the SQuAD evaluation, the T5 Base model is better than the T5 NewsQA.

V. Evaluation

We evaluated both the entire pipeline and some specific phases using a corpus especially created for the purposes of this paper. The corpus consists of texts appropriate for teaching English at the beginners level, and was obtained by extending a corpus created in a previous work [2].

As shown in figure 2, the corpus originally had the following proportion of question types: What: 45.3%, Who: 47.8%, When: 0.9%, Where: 5.0%, How: 0.3%. After extending it to evaluate the present work, the corpus resulted in a slightly more balanced corpus: What: 48%, Who: 38.5%, When: 2%, Where: 8.4%, How: 3.1%. Although there are still few When, Where and How questions, their proportion in the corpus increased significantly. The total number of question-answer pairs annotated in this corpus is 454 (in a total of 25 short texts).

[1] Suraj Patil. Question generation. https://github.com/patil-suraj/question_generation, 2020.

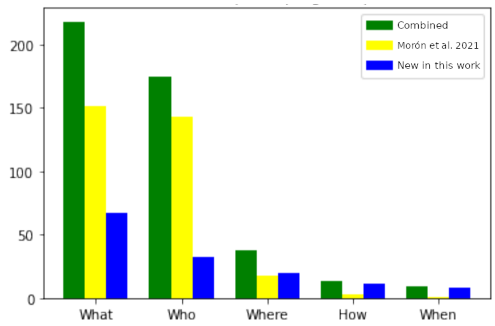

Fig. 2. Composition of our test dataset broken down by type of question. We show the datasaet by Morón et al. 2021 [2] and the contributions of the current work.

Table III shows the results of the three question generation models (third stage) on our corpus. We show results with and without the co-reference resolution step (first stage).

Model	Corefs.	BLEU	METEOR	ROUGE$_L$
T5 Base	No	30.8	42.7	68.3
T5 NewsQA	No	25.6	40.9	62.9
T5 SQuAD+NewsQA	No	18.8	37.8	53.6
T5 Base	Yes	35.1	43.0	67.9
T5 NewsQA	Yes	50.8	46.8	77.2
T5 SQuAD+NewsQA	Yes	35.5	44.1	69.9

TABLE III

COMPARISON OF THE TRAINED MODELS OVER OUR OWN TEST DATA. THE COLUMN "COREFS." INDICATES IF THE EXPERIMENT USES THE CO-REFERENCE RESOLUTION STEP.

As can be seen, co-reference resolution produces a significant improvement in the results. The T5-NewsQA model with co-reference resolution outperforms the other models for all the metrics. Additionally, all the results of our question generation models, even without the co-reference phase, outperform the results obtained on the SQuAD and NewsQA evaluation partitions when computed on our evaluation corpus. We believe this happens because the sentences in this corpus are very simple, intended to teach English to beginners.

Table IV shows the results of the complete pipeline (the four stages) using the best two models according to table III. The table considers the following configurations:

- **Pipeline 1**: Using the co-reference resolution module (stage 1) and the "T5 NewsQA" model (stage 3).
- **Pipeline 2**: Using the co-reference resolution module (stage 1) and the "T5 SQuAD+NewsQA" model (stage 3).

The table is split in two sections, according to two different evaluations we carried. First of all, we tried to evaluate the question-answer pairs generated by both pipelines using our test dataset as a gold standard. As can be seen in table IV, the results considered in this way are quite low both in terms of precision and recall. This could be happening because systems are generating incorrect outputs, but there is another possible explanation: it could be generating question-answer pairs that are correct, but were not considered in the test dataset. Some generated question-answer pairs could be variants of existing pairs in the dataset, and also the dataset could not be exhaustive enough and some generated pairs could just not be there.

Model		Pipeline 1	Pipeline 2
Total generated pairs		683	679
Comparison to our test dataset	Precision	0.31	0.25
	Recall	0.47	0.39
	F1	0.37	0.31
Analysis of generated pairs	OK	68.8%	68.3%
	Wrong	24.9%	25.5%
	Strange	6.3%	6.2%

TABLE IV

COMPARISON OF THE TWO PIPELINES FOR QUESTION-ANSWER GENERATION. PIPELINE 1 USES THE "T5 NEWSQA" MODEL, WHILE PIPELINE 2 USES THE "T5 SQUAD+NEWSQA" MODEL.

Because of this, we did a second stage of evaluation: we manually inspected all the generated question-answer pairs and tried to classify them in three categories: some of them were *OK* without any problems, some of them were completely *wrong*, and some of them were correct at first glance but could sound *strange* in English. The second part of table IV shows this evaluation. We can see that most of the questions (more than 68%) generated by both systems are OK (i.e. completely correct), and about a quarter of the questions could be considered incorrect.

Inspecting the different reasons why some of the question-answer pairs were incorrect, we found that, although the co-reference resolution module improved the quality of the results significantly (see table III), it was still far from perfect. On many occasions some of the candidate answers were just *"we"* or *"my"*, and the model struggled to come up with a suitable question that could match the answer.

Other wrong cases were due to errors in the SRL extraction module, which returned more than one sentence in the same span of text, such as *"her mom takes Maria to the dentist. Maria is scared"* or *"to see the sun. The kids like slides. The kids also like swings"*.

Finally, other types of errors were exclusively due to the question generation models. For example when using the candidate answer *"bored"*, a system generated *"What is the cat doing?"*, when an appropriate question could have been *"How is the cat?"* or perhaps *"What is the emotional state of the cat?"*. All these type of errors provide us with interesting information on ways to improve the system in the future.

VI. CONCLUSIONS

We developed a pipeline for the creation of educational activities with a question-answer format, that are generated from an English text selected by a teacher. Our approach is based on four stages: a pre-processing stage that, among other basic tasks, applies a co-reference resolution tool; an answer candidate selection stage, which is based on semantic role labeling; a question generation stage, which takes as input the text with the resolved co-references and returns a set of questions for each answer candidate; and a post-processing stage that adjusts the format of the generated questions.

We evaluated the question generation model in isolation in order to compare it with related work, using the SQuAD portion used as a benchmark. The model gives similar results to most of the previous work.

On the other hand, we evaluated the model on our own corpus, composed of simple English texts appropriate for ESL teaching at the beginner level. On this corpus, the best performing model is T5-NewsQA (T5 pre-trained with SQuAD with fine tuning using NewsQA), including the previous stage of co-reference resolution.

Finally, we evaluated the complete pipeline, which also includes the candidate answer selection stage. In this evaluation we analyzed the results manually and concluded that 68.8% of the question/answer pairs generated by T5-NewsQA are correct.

The developed tool is already functional and will be integrated into an existing platform of educational activities for teaching English. The platform includes an editing stage which will allow the teacher to correct any errors that may be generated.

The tool can be improved in different ways. It is possible to evaluate other language models to take as a starting point and extend the evaluation corpus to have a more accurate evaluation. An interesting future work is the elaboration of a corpus of questions and answers constituted by texts of the expected level of English, in order to carry out a new fine tuning to better adapt the model to the specific objective we are looking for.

References

[1] D. Litman, "Natural language processing for enhancing teaching and learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[2] M. Morón, J. Scocozza, L. Chiruzzo, and A. Rosá, "A tool for automatic question generation for teaching english to beginner students," in *2021 40th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2021, pp. 1–5.

[3] M. Heilman, "Automatic factual question generation from text," Ph.D. dissertation, Carnegie Mellon, 2011.

[4] R. Das, A. Ray, S. Mondal, and D. Das, "A rule based question generation framework to deal with simple and complex sentences," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 542–548.

[5] N. Le, T. Kojiri, and N. Pinkwart, *Automatic Question Generation for Educational Applications – The State of Art*. Springer, 2014.

[6] P. Khullar, K. Rachna, M. Hase, and M. Shrivastava, "Automatic question generation using relative pronouns and adverbs," in *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 153–158. [Online]. Available: https://www.aclweb.org/anthology/P18-3022

[7] M. Chinkina and D. Meurers, "Question generation for language learning: From ensuring texts are read to supporting learning," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 334–344. [Online]. Available: https://www.aclweb.org/anthology/W17-5038

[8] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," 2018.

[9] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," 2016.

[10] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1342–1352. [Online]. Available: https://www.aclweb.org/anthology/P17-1123

[11] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study," in *NLPCC*, 2017.

[12] X. Du and C. Cardie, "Harvesting paragraph-level question-answer pairs from Wikipedia," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1907–1917. [Online]. Available: https://aclanthology.org/P18-1177

[13] L. Song, Z. Wang, W. Hamza, Y. Zhang, and D. Gildea, "Leveraging context information for natural question generation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 569–574. [Online]. Available: https://aclanthology.org/N18-2090

[14] B. Liu, M. Zhao, D. Niu, K. Lai, Y. He, H. Wei, and Y. Xu, "Learning to generate questions by learningwhat not to generate," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1106–1118. [Online]. Available: https://doi.org/10.1145/3308558.3313737

[15] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, *Unified Language Model Pre-Training for Natural Language Understanding and Generation*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[16] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[19] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[21] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," *arXiv preprint arXiv:1803.07640*, 2018.

[22] P. R. Kingsbury and M. Palmer, "From treebank to propbank." in *LREC*. Citeseer, 2002, pp. 1989–1993.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the ACL*. Philadelphia, Pennsylvania, USA: ACL, 2002, pp. 311–318.

[25] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*. Ann Arbor, Michigan: ACL, Jun. 2005, pp. 65–72. [Online]. Available: https://aclanthology.org/W05-0909

[26] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: ACL, Jul. 2004, pp. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04-1013