

Ellipsis Resolution as Question Answering: An Evaluation

Rahul Aralikkatte¹, Matthew Lamm², Daniel Hardt³, and Anders Søgaard¹

¹University of Copenhagen, ²Stanford University

³Copenhagen Business School

Abstract

Most, if not all forms of ellipsis (e.g., ‘*so does Mary*’) are similar to reading comprehension questions (‘*what does Mary do*’), in that in order to resolve them, we need to identify an appropriate text span in the preceding discourse. Following this observation, we present an alternative approach for English ellipsis resolution relying on architectures developed for question answering (QA). We present both single-task models, and joint models trained on auxiliary QA and coreference resolution datasets, clearly outperforming the current state of the art for Sluice Ellipsis (from 70.00 to 86.01 F₁) and Verb Phrase Ellipsis (from 72.89 to 78.66 F₁).

1 Introduction

Ellipsis resolution is a hard, open problem in NLP, and an important source of error in machine translation, question answering, and dialogue understanding (Vicedo and Ferrández, 2000; Dzikovska et al., 2009; Chung and Gildea, 2010; Macketanz et al., 2018; Petr Bach Hansen and Sgaard, 2020). There are no large annotated text corpora for this phenomenon, even for English, and we only have annotations for a subset of the known ellipsis constructions. Since annotation is expensive and cumbersome, any synergies with existing NLP tasks could be useful and enable us to leverage auxiliary data when learning models for ellipsis resolution.

This paper presents a simple yet strong approach to ellipsis resolution based on a straightforward observation, depicted in Figure 1, that ellipsis resolution can be converted to a QA problem. Ellipsis and questions put in focus *referentially dependent* expressions (Carlson, 2006), or free variables (Partee, 1978), that need to be resolved in order to comprehend the discourse. For similar observations about different tasks, see McCann et al. (2018) and Gardner et al. (2019).

This straightforward observation leads us to

Sluice Ellipsis

Context: ... But the way things are structured now you have to set aside your ego to make things happen. **The whole thing worked out.** I don't know **how**, but it did. Both sides had to work to make it happen ...

Question: I don't know how, but it did.

Answer: The whole thing worked out

Verb Phrase Ellipsis

Context: ... It has to be considered as an additional risk for the investor," said Gary P. Smaby of Smaby Group Inc., Minneapolis. "Cray Computer will be a concept stock," he said. "You either **believe Seymour can do it** again or you **don't** ...

Question: You either believe Seymour can do it again or you don't.

Answer: believe Seymour can do it again

Figure 1: Examples of Sluice Ellipsis and Verb Phrase Ellipsis, represented as “questions” about their associated contexts. Wh-phrases and auxiliary verbs are marked in red and elided phrases are marked in blue.

suggest treating different forms of ellipsis resolution – and later, as an auxiliary task, coreference resolution – as a QA problem, and to apply state-of-the-art architectures for QA to ellipsis resolution tasks, as well as to experiment with using training data for QA and coreference resolution to improve our new ellipsis resolution models.

Contributions We cast ellipsis as a QA problem, enabling us to induce models for it using neural architectures originally developed for QA. Applying these architectures out of the box enables us to establish strong results¹ for ellipsis resolution tasks, improving significantly over previous work. Using the same architecture for the different ellipsis resolution tasks, as well as for QA and coreference resolution, enables us to explore syn-

¹Though we report state-of-the-art results for both sluice and verb phrase ellipsis, we consider these models as strong baselines for future research as they are obtained purely using existing methods.

ergies between the tasks, and we show that training joint models on these tasks leads to even better performance.

2 Methodology

In this section, we briefly describe the various datasets used for training, and explain how they are converted into QA format. We then move on to the choice of model architectures and the reasoning behind their selection.

Sluice Ellipsis For training and evaluation of Sluice Ellipsis resolution models, we use the corpus introduced by [Anand and McCloskey \(2015\)](#), which contains 3,103 annotated examples of embedded sluices, collected from the New York Times section of the English Gigaword corpus. Since the annotators were free to paraphrase the antecedent, in some cases, a string match on the context does not return antecedent span indices. To ensure a fair comparison, we follow previous work ([Rønning et al., 2018](#)), which is also the current state-of-the-art, in ignoring these instances, and use their split for training, development and testing.

Verb Phrase Ellipsis [Bos and Spenader \(2011\)](#) provide Verb Phrase (VP) Ellipsis annotations for the WSJ part of the Penn Treebank. All 25 sections were annotated, and we follow them in using sections 0-19 for training, and 20-24 for testing. We further hold out sections 18-19 from the training data for development. This also enables to us compare our results directly with the current state-of-the-art for VP Ellipsis ([Zhang et al., 2019](#)).

Coreference Resolution For coreference resolution, which we use as an auxiliary task, we train and evaluate on two corpora: (i) the English portion of the OntoNotes 5.0² corpus with the standard data split used in the CoNLL-2012 shared task ([Pradhan et al., 2012](#)), and (ii) the WikiCoref corpus ([Ghaddar and Langlais, 2016](#)), which contains annotations of 30 documents from the English Wikipedia. From this dataset, we use 22 documents for training, 4 documents for development, and 4 for testing.

QA We also use SQuAD v1.1 ([Rajpurkar et al., 2016](#)) as an auxiliary reading comprehension dataset.

²<https://catalog.ldc.upenn.edu/LDC2013T19>

Task	Train	Dev	Test	ACL
ELLIPSIS				
Sluice Ellipsis	1.4k	480	992	351
VP Ellipsis	264	20	78	984
AUXILIARY				
OntoNotes	153k	18.8k	19.5k	463
WikiCoref	5.6k	630	638	2.2k
SQuAD	87.6k	10.6k	-	117

Table 1: QA pair counts and average context lengths (ACL) for different datasets, after conversion

Data Conversion For converting the various datasets into the QA format of `<context, question, answer>` triples, we perform a simple restructuring as shown in Figure 1. We consider the entire document as the context; the sentence in which the ellipsis/mention is present becomes the question, and the antecedent/entity becomes the answer. In case of coreference resolution, where a single sentence can have n mentions, we create n questions where every question is the same sentence with a different mention $i \in \{1 \dots n\}$ marked for resolution with `<ref>` and `</ref>` tags. Table 1 shows the number of QA pairs created from each dataset and the average number of words in their contexts.

QA Architectures Generally, QA models have two main components: (i) an encoder module which learns to represent the question and its context, and (ii) a span selection module which predicts the start and end span indices of the answer if it is present in the context. In this work, we present experiments with three diverse models which take entirely different approaches to build the encoder module: (i) DrQA ([Chen et al., 2017](#)), with an LSTM encoder, (ii) QANet ([Yu et al., 2018](#)), with a CNN encoder, and (iii) BERT ([Devlin et al., 2019](#)), with a (pretrained) transformer encoder. We use the three different models to show that the between-task synergies are relatively robust across architectures; but one architecture (BERT) is clearly superior to the others and will be the standard baseline we propose for future research.³

³Note that there are many differences between these architectures; not only the encoder networks. The number of parameters differ, and BERT is pre-trained on large volumes of data. Our purpose here is not comparing strategies, but simply showing that synergies can be seen across all architectures. For more details, see Appendix B.

TASK	SOTA	SINGLE TASK			JOINT		
		DRQA	QANET	BERT	DRQA	QANET	BERT
Sluice Ellipsis	70.00 (Rønning et al., 2018)	77.48	75.70	<u>85.10</u>	80.17	77.11	86.01
VP Ellipsis	72.89 (Zhang et al., 2019)	62.86	1.93	76.42	63.54	22.49	<u>78.66</u>

Table 2: Ellipsis resolution scores are token-level F_1 . Bold-faced results are better than the previous state-of-the-art; underlined results are the new state-of-the-art. When evaluated, our best joint architecture scores 72.31 on OntoNotes and 65.30 on WikiCoref (macro-averages of MUC, B^3 , and $CEAF_{\phi_4}$ scores). See Appendix C.2 for why these numbers are not directly comparable to previously reported coreference resolution results in literature.

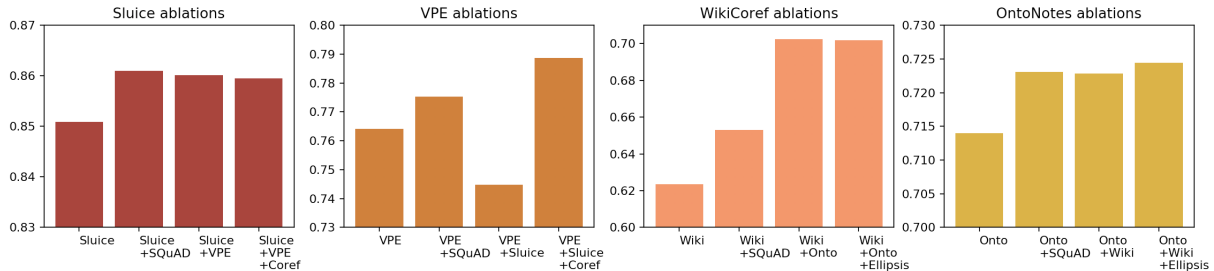


Figure 2: Dataset ablations (F_1)

3 Experiments & Results

We conduct two sets of experiments: (i) the SINGLE-TASK experiments, in which we train and evaluate separate models for the two ellipsis resolution tasks; and (ii) the JOINT modelling experiments, where we train on the best possible combination of ellipsis resolution, coreference resolution and QA data, as determined on the validation set. The results can be seen in Table 2.⁴

Single-Task Setup The SINGLE-TASK DrQA model improves the state-of-the-art on sluice ellipsis by 7.48 F_1 . The SINGLE-TASK QANet model also improves the state-of-the-art on sluice ellipsis by 5.7 F_1 , but fails to learn anything meaningful for VP ellipsis. We hypothesise this is due to the fact that 264 training examples are not enough to train the model’s large stack of encoder blocks from scratch.

The SINGLE-TASK BERT model achieves state-of-the-art results in both the ellipsis datasets with absolute error reductions of 50.33% (Sluice Ellipsis) and 13.02% (VP Ellipsis). Interestingly, it also achieves a 17.10% error reduction over the best previously reported results on WikiCoref, but see Appendix C.2 for why such a direct comparison of numbers is not entirely fair.

Joint Setup The JOINT models always perform on-par with, or better than the SINGLE-TASK

models. In this setup, the BERT models beat the previous state-of-the-art for both Sluice and VP Ellipsis with 53.37% and 21.28% absolute error reductions respectively.

4 Dataset ablations

We determine the best task combinations on held-out validation data for each ellipsis resolution task.⁵ For Sluice Ellipsis, the best results are obtained by training the models on a combination of Sluice and VP Ellipsis data. For VP Ellipsis, the best performance is attained when the models are trained with a combination of all datasets. When training a model for a particular task, we sample auxiliary data from other datasets to match the size of the main task’s dataset. For each dataset, the variations in its F_1 scores of the best performing architecture when combined with other datasets are shown in Figure 2. The most interesting findings from these ablations are mentioned below.

When the two ellipsis datasets are combined, the overall performance of the models increase for both tasks by around 1% each. This shows that the two types of ellipsis are similar, and that when learning ellipsis resolution models, there is considerable synergy between the two resources. If we add subsampled coreference data when training these models, the Verb Phrase Ellipsis models gain up to 2.9%. One possible explanation

⁴The reported results are the average of three independent runs with different random seeds.

⁵These ablations are performed on the best performing (BERT) model.

<p>Then at 10:15, the Dow suddenly started to rebound, and when it shot upward it did so even faster than the early-morning fall.</p> <p>Gold shot upward VPE_s shot upward VPE_j it shot upward</p> <p>Example (a)</p>	<p>A 190-point drop isn't likely to make much of a dent; multiply that a few times over, though, and it will.</p> <p>Gold make much of a dent VPE_s make much of a dent; multiply that a few times over VPE_j A 190-point drop isn't likely to make much of a dent</p> <p>Example (b)</p>	<p>Then the whole thing will start to collapse, just as it did in the 1970s, and the ghosts and banshees will be howling through the place turning people's hair white.</p> <p>Gold collapse VPE_s go to war to stop anyone from trying to grab Iran. But that ghost wouldn't settle for words, he wanted money and people VPE_j collapse</p> <p>Example (c)</p>
--	--	---

Figure 3: Selected gold and predicted antecedent spans from SINGLE-TASK Verb Phrase Ellipsis (VPE_s in figure) and JOINT Verb Phrase Ellipsis (VPE_j in figure) models.

could be more similarities between noun phrases and verb phrases, than between noun phrases and the sentences that are elided in Sluice Ellipsis resolution.

5 Error Analysis

We now look at some errors made by our best performing models. First, we compare the errors made by our SINGLE-TASK and JOINT Sluice Ellipsis resolution models before moving on to VP Ellipsis.⁶

Sluice Ellipsis The JOINT Sluice Ellipsis results improve modestly over the SINGLE-TASK Sluice Ellipsis results. This is noteworthy, since the added VP Ellipsis data is quite small compared to the size of the sluice data. These models consistently select an antecedent of the right syntactic form, which is normally a complete sentence. Many of the errors consist of empty outputs: SINGLE-TASK Sluice Ellipsis produces 58 empty outputs, while JOINT Sluice Ellipsis produces 63. Another source of error is discontinuous antecedents. It is not unusual for the gold antecedent to be a discontinuous span (Donecker, 1996), but our models are not permitted to produce such antecedents, so these cases will always be a source of error.

All the systems have problems when the antecedent follows the ellipsis, as in the following example: *I don't know why, but they seem to need a story*. We also compared the right and left periphery scores of sluices, and found better results predicting the right periphery: for SINGLE-TASK Sluice Ellipsis, there were 678 matches on the left edge, and 733 on the right edge; for JOINT Sluice

Ellipsis, there were 703 left matches and 734 right matches.

Verb Phrase Ellipsis The SINGLE-TASK VP models trained with just VP Ellipsis data improves on the current state of the art, and further improvement is observed when trained on auxiliary data, especially the Sluice Ellipsis resolution dataset. While the JOINT VP Ellipsis model is generally better than the SINGLE-TASK model, joint training with Sluice Ellipsis resolution data also seems to introduce unfortunate biases. While the SINGLE-TASK model always selects antecedents of the right syntactic form, i.e., verb phrases, the JOINT model may select sentential antecedents. See examples in Figure 3.

In Example (a), the JOINT VP model incorrectly includes the subject *it*, presumably because the sluice data includes complete sentences as antecedents. Similarly in Example (b) – though the SINGLE-TASK model correctly chooses an antecedent beginning with the verb *make*, it continues with additional material that does not form a coherent antecedent. The JOINT result is also incorrect, but note that it consists of the complete sentence containing the correct VP antecedent. Example (b) presents the advantages and disadvantages of the joint ellipsis training data. While the two types of ellipsis require antecedents of different forms, they have similar requirements in terms of where in the context the antecedent is to be found. Example (c) further supports this point. Here the JOINT result is perfect, while the SINGLE-TASK result finds an antecedent that is in the wrong part of the discourse. The SINGLE-TASK model is slightly better with left periphery matches than right: we found 58 left and 55 right matches. This is reversed with the JOINT model, with 54 left and 60 right matches.

⁶We also briefly discuss how coreference resolution benefits from synergies with ellipsis in Appendix C.1.

6 Related Work

We are not the first to use question answering to redefine a set of tasks. Recently, He et al. (2015) showed that semantic role labeling annotations could be solicited by asking simple questions that implicitly target predicate-argument relations in a sentence. Parallel to our work, Hou (2020) cast bridging anaphora resolution as question answering based on context. Wu et al. (2020) and Li et al. (2020) also reformulate coreference resolution and named entity recognition as QA. In the realm of re-framing relation extraction as a QA problem, Levy et al. (2017) and Abdou et al. (2019) create monolingual and multilingual template based QA datasets respectively, which yield relation extraction models which were better at generalizing in the zero-shot setting. Extending this idea, McCann et al. (2018) introduced the DecaNLP challenge, which casts 10 core tasks in NLP as question-answering problems. Similar to our work, their architecture jointly learns across all of these tasks. DecaNLP includes pronoun resolution, a subset of coreference resolution, but it does so only on a small, hand-crafted dataset; it does not address ellipsis.

Limitations of our approach One limitation of our approach is that, like most previous work, we assume ellipsis and coreference resolution amount to finding antecedent spans that corefer with the target mention. This is not always the case; the elided material can: (i) have extra-linguistic antecedents, and (ii) refer to something that is contextually implied.

7 Conclusion

We present strong models for Sluice and Verb Phrase ellipsis resolution problems, by reformulating them as machine reading comprehension problems, significantly outperforming the previously best reported results. We also empirically show that training these models jointly and with auxiliary data from coreference resolution and question-answering further improves their performance. Our code is publicly available at <https://github.com/rahular/ellipsis-baselines>.

8 Acknowledgements

We thank the reviewers for their valuable feedback. Rahul Aralikkatte and Anders Søgaard are funded by a Google Focused Research Award.

References

- Mostafa Abdou, Cezar Sas, Rahul Aralikkatte, Isabelle Augenstein, and Anders Søgaard. 2019. [X-WikiRE: A large, multilingual resource for relation extraction as machine comprehension](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 265–274, Hong Kong, China. Association for Computational Linguistics.
- Pranav Anand and Jim McCloskey. 2015. [Annotating the implicit content of sluices](#). In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 178–187, Denver, Colorado, USA. Association for Computational Linguistics.
- Rajendra Banjade, Vasile Rus, and Nopal B. Niraula. 2015. Using an implicit method for coreference resolution and ellipsis handling in automatic student answer assessment. In *FLAIRS*.
- Johan Bos and Jennifer Spender. 2011. [An annotated corpus for the analysis of vp ellipsis](#). *Lang. Resour. Eval.*, 45(4):463–494.
- Greg Carlson. 2006. Anaphora. In *Encyclopedia of Cognitive Science*. Wiley.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Tagyoung Chung and Daniel Gildea. 2010. [Effects of empty categories on machine translation](#). In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 636–645, Cambridge, MA. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Paul G. Donecker. 1996. [Subdeletion in verb phrase ellipsis](#). In *34th Annual Meeting of the Association for Computational Linguistics*, pages 348–350, Santa Cruz, California, USA. Association for Computational Linguistics.
- Myroslava Dzikovska, Charles Callaway, Elaine Farrow, Johanna Moore, Natalie Steinhauser, and Gwendolyn Campbell. 2009. [Dealing with interpretation errors in tutorial dialogue](#). In *Proceedings of the SIGDIAL 2009 Conference*, pages 38–45, London, UK. Association for Computational Linguistics.

- Paul Elbourne. 2013. *Definite descriptions*, volume 1. Oxford Studies in Semantics.
- Matt Gardner, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. [Question answering is a format; when is it useful?](#)
- Abbas Ghaddar and Philippe Langlais. 2016. Wikicoref: An English coreference-annotated corpus of Wikipedia articles. In *LREC*.
- Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. [Question-answer driven semantic role labeling: Using natural language to annotate natural language](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Comput.*, 9(8):1735–1780.
- Yufang Hou. 2020. [Bridging anaphora resolution as question answering](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1428–1438, Online. Association for Computational Linguistics.
- Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. [Higher-order coreference resolution with coarse-to-fine inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692. Association for Computational Linguistics.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342. Association for Computational Linguistics.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. [A unified MRC framework for named entity recognition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.
- Chuan-Jie Lin, Chien-Wei Pao, Yen-Heng Chen, Ching-Ting Liu, and Hui-Huang Hsu. 2016. [Ellipsis and coreference resolution in a computerized virtual patient dialogue system](#). *Journal of Medical Systems*, 40.
- Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. [Fine-grained evaluation of German-English machine translation based on a test suite](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 578–587, Belgium, Brussels. Association for Computational Linguistics.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. [The natural language deathlon: Multitask learning as question answering](#). *CoRR*, abs/1806.08730.
- Jason Merchant. 2001. *The syntax of silence: Sluicing, islands, and the theory of ellipsis*. Oxford University Press on Demand.
- Barbara H. Partee. 1978. [Bound variables and other anaphors](#). In *Theoretical Issues in Natural Language Processing-2*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Victor Petrn Bach Hansen and Anders Sgaard. 2020. [What do you mean why?: Resolving sluices in conversations](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7887–7894.
- Paul M. Postal. 1966. On so-called pronouns in english. In F. Dinneen, editor, *Nineteenth Monograph on Language and Linguistics*. Georgetown University Press.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. [Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes](#). In *Joint Conference on EMNLP and CoNLL - Shared Task, CoNLL '12*, pages 1–40, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Ola Rønning, Daniel Hardt, and Anders Søgaard. 2018. [Sluice resolution without hand-crafted features over brittle syntax trees](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, page 236241. Association for Computational Linguistics.
- José L. Vicedo and Antonio Ferrández. 2000. [Importance of pronominal anaphora resolution in question answering systems](#). In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 555–562, Hong Kong. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. [CorefQA: Coreference resolution as query-based span prediction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. [Qanet: Combining local convolution with global self-attention for reading comprehension](#). In *ICLR (Poster)*. OpenReview.net.

Weinan Zhang, Yue Zhang, Yuanxing Liu, Donglin Di, and Ting Liu. 2019. [A neural network approach to verb phrase ellipsis resolution](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:7468–7475.

A Similarity between Ellipsis and Coreference Resolution

Linguists have long pointed out deep links among different forms of ellipsis, as well as between ellipsis and pronominal anaphora. For example, [Merchant \(2001\)](#) presents a unified account of ellipsis phenomena within a minimalist syntactic framework, and theorists such as [Postal \(1966\)](#) and [Elbourne \(2013\)](#) go so far as to argue that pronouns are also elliptical forms. The exact nature of the connections between ellipsis and anaphoric constructions remains a subject of controversy among linguists. However it is clear that there are rooted connections, and in our view these connections represent potential areas to be exploited with forms of knowledge transfer among datasets of different types.

Typically in NLP, ellipsis and coreference have been treated as distinct tasks. Possible exceptions

include [Lin et al. \(2016\)](#), who present a rule-based, feature-rich system for handling ellipsis and coreference in Chinese medical dialogues, but the synergy between the two subsystems is limited; and [Banjade et al. \(2015\)](#), who reduce ellipsis and coreference to problems of alignment to an auxiliary text implicitly describing the universe of the dialogue in question.

B QA Models

We briefly describe the architectures of the QA models below. All experiments are conducted on a single 12 GB GPU. For all models, we use the hyperparameter values recommended in their respective papers.

DrQA The Document Reader component of DrQA consists of a context and a question encoder followed by two span prediction classifiers. The context encoder is a multi-layer bi-directional LSTM ([Hochreiter and Schmidhuber, 1997](#)) which takes in word embeddings ([Pennington et al., 2014](#), GloVe), similarity based features (whether the token appears in the question in its original, lowercase or lemma form), and other token level features (positional tags, named entities and term frequency) as input. The concatenation of each layer’s hidden units is used as the context vector. The question encoder is another LSTM which takes word embeddings as input and combines the resulting hidden units using a simple attention mechanism to form the question vector. A bilinear term which captures the similarities between context and question vectors is used to combine the two vectors and the resulting vector used as input to the span prediction classifiers. The two classifiers predict the start and the end span respectively and are trained independently.

QANet In QANet, each encoder layer is a stack of depthwise separable convolutions followed by a multi-head self-attention mechanism placed inside a residual block. Initially, words in the context and question are embedded using a combination of GloVe and character embeddings. They are then contextualized individually with an encoder block. The representations are then passed through a context-query attention layer to obtain a combined representation of the context and question. This is further passed through three encoding blocks before feeding it into a classifier for predicting the answer spans.

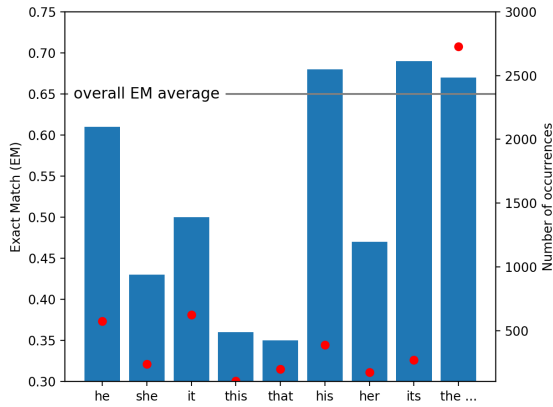


Figure 4: Exact match percentage (bars) and number of occurrences (dots) of referential forms in OntoNotes

BERT We use the pre-trained BERT_{BASE} uncased model to encode questions and their contexts. It has 12 Transformer blocks, 12 self-attention heads, and a hidden size of 768. Word piece tokenization (Wu et al., 2016) is performed, both on the context paragraph and the question. The boundaries of the two sequences are marked by dummy symbols. The context and the question are joined with a *[SEP]* token in between, and the *[CLS]* token is prepended at the beginning to form the input. The representation of the *[CLS]* token is fed into a single-layer MLP with 2 outputs which is used to predict the span indices.⁷

C Coreference Resolution

In this section, we analyse the best performing coreference models and discuss why they cannot be compared with other works in literature.

C.1 Error Analysis

The JOINT OntoNotes model improves a little over the SINGLE-TASK counterpart. Here we examine specific referential forms in OntoNotes (WikiCoref has similar traits), as shown in Figure 4. In general, performance is better on frequent pronouns – e.g., ‘he’ over ‘she’, ‘this’ and ‘that’. An exception to this is that ‘it’ is less accurate, but more frequent than ‘he’. It is notable that the possessive pronouns (‘his’, ‘her’, ‘its’) are all more accurate than their nominative counterparts (‘he’, ‘she’, ‘it’), perhaps because they tend to have a closer connection to their antecedents. Overall, the single-word referential forms are less accurate than multiple-word forms. For example, definite

descriptions (forms beginning with ‘the’) are more accurate than any of the single-word forms, with the exception of ‘its’. We speculate that multi-word forms provide more specific information, thus limiting the set of potential antecedents. It is also interesting to break down error by the grammatical gender of the pronouns. Male pronouns generally tend to be more accurate than their female counterparts. Antecedents of ‘he’ and ‘his’ are matched 20% more frequently than for ‘she’ and ‘her’. This is probably due to an unfortunate bias in OntoNotes, where female pronouns are 50% rarer than male pronouns.

C.2 Result Comparability

Converting coreference into QA fundamentally changes the coreference resolution problem: It, on the one hand, makes the coreference resolution problem harder, in that we require the identification of a specific antecedent span, rather than any mention in the entity chain; on the other hand, the problem becomes easier by providing the bracketing of the mention that needs to be resolved. Due to these differences, it is not possible to directly compare our results with others in literature. For analysis, to make our results more comparable with Lee et al. (2018), we provided their model with the bracketing of the mentions and considered the first mention to be the antecedent. This way we can reinterpret their clusters as question-answer pairs and do not penalize them for mention bracketing errors, only considering pairs where they correctly identify mentions. Note this gives their model an advantage over ours, as their model considers multiple sources of evidence for inferring the coreference links, and gets to pick the subset of data on which the models are compared. On OntoNotes, in this setting, and after pruning around 7,358 mentions Lee et al. (2018) bracketed wrongly, their new average F₁ score is 75.9. Our performance on the same subset of the data is 72.1. Upon manual inspection, we see the model in Lee et al. (2018) has a strong bias favoring nominal antecedents, whereas our model is more likely to predict clausal antecedents. On WikiCoref, our model remains better than the previous state of the art by some margin, with an F₁ of 69.2 over 43.6.

⁷We use the implementation detailed in Wolf et al. (2020).