Sequence to Sequence Coreference Resolution

Gorka Urbizu
Elhuyar Fundation
gurbizu@elhuyar.eus

Ander Soraluze and **Olatz Arregi**
HiTZ Center, Ixa NLP group,
University of the Basque Country
{ander.soraluze, olatz.arregi}@ehu.eus

Abstract

Until recently, coreference resolution has been a critical task on the pipeline of any NLP task involving deep language understanding, such as machine translation, chatbots, summarization or sentiment analysis. However, nowadays, those end tasks are learned end-to-end by deep neural networks without adding any explicit knowledge about coreference. Thus, coreference resolution is used less in the training of other NLP tasks or trending pretrained language models. In this paper we present a new approach to face coreference resolution as a sequence to sequence task based on the Transformer architecture. This approach is simple and universal, compatible with any language or dataset (regardless of singletons) and easier to integrate with current language models architectures. We test it on the ARRAU corpus, where we get 65.6 F1 CoNLL. We see this approach not as a final goal, but a means to pretrain sequence to sequence language models (T5) on coreference resolution.

1 Introduction

Coreference resolution is a Natural Language Processing (NLP) task which consists on identifying and clustering all the expressions referring to the same real-world entity in a text. NLP tasks that include language understanding such as text summarisation (Steinberger et al., 2016; Kopeć, 2019), chatbots (Agrawal et al., 2017; Zhu et al., 2018), sentiment analysis (Krishna et al., 2017) or machine translation (Werlen and Popescu-Belis, 2017; Ohtani et al., 2019) can benefit from coreference resolution. And until recently, coreference resolution has been a critical task on the pipelines of those systems.

However, with the recent rising trend of building end-to-end deep neural networks, for any NLP task where the data available in that language or domain is huge, current models are able to learn the end task without any explicit training on coreference resolution. This is even more evident in the case of the huge unsupervisedly pretrained language models (LM) that are already able to resolve coreference (Clark et al., 2019; Tenney et al., 2019), as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2019), or GPT3 (Brown et al., 2020) which are used to boost results on any downstream task.

Those pretrained language models have also improved notably the results obtained at coreference resolution. Combining the SotA neural coreference resolution system (Lee et al., 2017) at the time with pretrained language models (ELMo, BERT, SpanBERT) improves results by a large margin.

Despite coreference resolution was already useful in NLP end tasks before the irruption of deep learning in NLP, and getting very significant improvements on the results with it, nowadays most of the tasks that require deep language understanding, are approached without having coreference resolution in mind.

Src:	Even	the	smallest	person	can	change	the	course	of	history	.
Trg:	(0	_	_	0)	_	_	(1	_	_	(2)|1)	_

Table 1: Example of sequence to sequence approach for coreference resolution.

Proceedings of the 3rd Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2020), pages 39–46,
Barcelona, Spain (online), December 12, 2020.

In this paper, we introduce a new approach to solve coreference resolution as a sequence to sequence task (as shown in Table 1) using a Transformer (Vaswani et al., 2017), that opens a path towards unifiying the approaches used in coreference resolution with the trending pretrained LMs and other NLP tasks, while simplifying the neural architecture used for coreference resolution.

We test our approach on the English ARRAU corpus (Uryupina et al., 2020), which includes singletons. We train our model on coreference resolution as a sequence to sequence task, where the neural network learns to produce the coreference relations as output from the raw text in the source.

In the following Section 2 we review the state of the art of the field. In Section 3 we describe how we approached coreference resolution as a sequence to sequence task, we present the neural architecture and corpora we used. In Section 4 we report our results, and lastly, we present our conlusions and future work in Section 5.

2 State of the Art

The SotA for English coreference resolution, improved a lot since the revolution of deep learning in NLP. The first end-to-end neural model (Lee et al., 2017) obtained big improvements over previous models. Since then, pretrained LMs improved a lot those results; adding ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and SpanBert (Joshi et al., 2020) to the model, improved by a large margins the SotA at the moment (Lee et al., 2018; Kantor and Globerson, 2019; Joshi et al., 2019; Joshi et al., 2020).

Furthermore, we would like to underline different approaches as reinforcement learning (Fei et al., 2019) and neural MCDM and fuzzy weighting techniques (Hourali et al., 2020), which improved results.

There have been only two works which already have tried to combine language models and coreference resolution at training. In the first one, T5 (Raffel et al., 2019), they use coreference resolution among other tasks to train a neural language model on text to text, but the coreference task is approached as a simple binary mention-pair task, which does not reflect all the advances done at resolving coreference. In the second one, CorefQA (Wu et al., 2020), they adress coreference resolution as query-based span prediction for which they convert coreference resolution into a QA task, where the model has to find the coreferential mentions in the text. Although they get the best results obtained to this day, their approach still uses a windowing technique of length 512, and needs to create questions automatically from the text.

Models	F1
(Lee et al., 2017)	68.8
(Lee et al., 2018)	73.0
(Fei et al., 2019)	73.8
(Kantor and Globerson, 2019)	76.6
(Joshi et al., 2019)	77.1
(Joshi et al., 2020)	79.6
(Hourali et al., 2020)	80.0
(Wu et al., 2020)	83.1

Table 2: The state of the art for English coreference resolution: F1 scores at CoNLL metric, for Ontonotes/CoNLL-2012 dataset.

We should keep in mind that, apart of the well studied English language, there are lots of other less researched languages. Yet we already have neural models for some of those languages: Polish (Nitoń et al., 2018), Japanese (Shibata and Kurohashi, 2018), French (Grobol, 2019), Basque (Urbizu et al., 2019), Telegu (Annam et al., 2019), Russian (Sboev et al., 2020) Persian (Sahlani et al., 2020) and cross-linguals (Cruz et al., 2018; Kundu et al., 2018) with varied results depending on corpus sizes and architectures.

3 Sequence to Sequence Coreference Resolution

Coreference resolution has been historically divided in two subtasks. The first one is mention detection, where posible candidates for a mention are located in the text. The second one would be to find those which have coreferential relations, among the mentions. This second task has been approached as a clustering problem, where mention-pair models evolved into entity-mention models, and their respectives ranking models. Some of this approaches have issues with making the correct global decisions, and those who handle this more appropriately, have higher computational cost. In the following subsection, we present our approach, which solves these two subtasks at once in a simpler way.

3.1 Our Approach

There are many ways to annotate or indicate coreference relations on a text, such as using 2 columns, which was used on the Ontonotes corpus (Pradhan et al., 2007) for the CONLL task (Pradhan et al., 2011; Pradhan et al., 2012). On the left we have the raw text word by word, and on the right, the coreference relations expressed in a parenthetical structure, were parenthesis are used to delimitate mentions, and numbers to refer the coreference clusters that the mentions belong.

Text:	Coreference:
you	(0)
love	_
me	(1)

Table 3: Two column annotation.

Source:	You	love	me
Target:	(0)	_	(1)

Table 4: Sequence to sequence task.

This annotation system shows that the task is similar to sequence-labeling tasks, where the labels of the second row are not discrete. To handle this problem, we propose a sequence to sequence approach. In source we would have the raw text, and in the target, the coreference annotation corresponding to the source text in the parenthetical structure.

To make the task easier to learn, as there are many equivalent ways to represent the same coreference relations, we rewrite all the numbers referring to coreference clusters in the training dataset, with ascendent numbers starting from 0, from left to right, keeping the coreference relations.

3.2 Transformer Model

We choose the architecture of Transformer, as it gives good results for many sequence to sequence tasks. Although keeping source and target sequences of the same length helps the model to create the outputs of the correct length, this creates the problem of huge vocabularies in source and target, which makes training the model harder, and more memory consuming.

To solve this issue, we use fixed vocabularies on source and target sequences. On source, we use BPE (Bojanowski et al., 2017) to segment words in subword units, with which we get a small closed vocabulary of 16K tokens. On target, we divide the labels of coreference resolution which contains more than one coreference relation within it, so that we avoid conplex labels, as (8)|122)|68)|128), which are hard to learn correctly: (8) | 122) | 68) | 128). Doing this, we decrease the size of the target vocabulary significantly (1.7K).

Src:	Even	the	small@@	est	person	can	change	the	course	of	history	.	
Trg:	(0	_	_	0)	_	_	(1	_	_	(2)	|	1)	_

Table 5: Example of source and target sequences.

As we can see in the example above, the aligment that we got previously is gone, so the model will have to learn to align source and target tokens, which a Transformer should do easily, as seen in tasks such as machine translation with this architecture. Furthermore, with those changes the source and target vocabularies sizes decrease a lot, making easier to understand the text and produce correct target tokens.

We do not use any pretrained word embeddings or LMs, or any other linguistic, distance or speaker features. We have choosen fairseq implementation of the Transformer (Ott et al., 2019) with standard hyperparameters. We set the max length of the source and target sequences at 1024. As coreference resolution is a document level task, it might happen that the document that we want to process has more than 1024 tokens in source or target after applying BPE and labels division. To handle that, a model with longer sequences should be trained (increasing significantly memory requirements), or a windowing strategy could be used. But we do not try any of this here, to keep computational costs low[1].

3.3 Datasets

We tested our approach on the ARRAU corpus (Uryupina et al., 2020), an English dataset which includes singletons. They had been ignored due to the division on mention detection and clustering tasks, and the specific corpora made for the second one. We train our Transformer model just to carry out both tasks at once. We used all coreference relations of the dataset. The corpus has 350K words, and its already divided on train, dev and test subsets.

As we do not add any pretrained word embeddings or any LMs to the model, the ARRAU corpus is not big enough to learn the task of language understanding in the encoder part and it has a limited vocabulary in the training. Thus, we used an auxiliary corpus for the training. We chose PreCo corpus, which is an English coreference corpus of over 10M words, which also includes singletons (Chen et al., 2018). Both datasets were converted to the mentioned two column format from their respective enriched annotations.

3.4 Data Augmentation

We used data augmentation to increase the amount of training instances. For this purpose, we took all the combinations of consecutive sentences for the training. Given the document $S_A - S_Z$, where S is a sentence: S_A , S_A-S_B, ..., S_A-S_B-S_C-...-S_Z; S_B, S_B-S_C, ... S_B-S_C-S_D-...-S_Z; ...; S_Y, S_Y-S_Z; S_Z.

With this technique, we do not improve much the dataset for source sequences, as it would be the same sentences repeated in different lengths. However, the repeated parts of the sequences in the source, would have their coreference relations represented by different numbers in the target sequences:

S_A-S_B-S_C Src:	You	love	cats	.	I	love	cats	.	My	dog	hates	cats	.
S_A-S_B-S_C Trg:	(0)	_	(1)	_	(2)	_	(1)	_	(3 | (2)	3)	_	(1)	_
S_B-S_C Src:					I	love	cats	.	My	dog	hates	cats	.
S_B-S_C Trg:					(0)	_	(1)	_	(2 | (0)	2)	_	(1)	_
S_C Src:									My	dog	hates	cats	.
S_C Trg:									(0 | (1)	0)	_	(2)	_

Table 6: Training sequences after data augmentation, and its effect on the target cluster numbers.

Furthermore, having sequences of a single sentence in the training, makes the beginning of the learning process easier. Later, the model will be able to learn to resolve coreference for whole documents at once.

3.5 Post-processing

Once we get the output prediction sequences, we need to post-process a bit the output with the 3 following processes. First, we correct the unclosed (or unopened) patenthesis or mentions, deleting them. Then, we group the different coreference relations referring to the same token again (just removing the space between each of the | in the output). Finally, we correct the length of the output sequence, removing tokens, or adding extra "_" tokens at the end until it matches the length of the source text. We can see the changes made to the predicted sequence at post-procesing in the following example:

Src:	Even	the	small@@	est	person	can	change	the	course	of	history	.		
Trg:	(0	_	_		0)	_	_	(1	_	_	(2)	|	1)	_
Pred:	(0	_	_		0)	_	_	(1	(2	_	(3)	|	1)	
Post:	(0	_	_		0)	_	_	(1	_	_	**(3)|1)**	_		

Table 7: Example of the post-procesing applied to the predicted sequences.

[1]We trained the model on a single Nvidia Rtx 2080Ti GPU (11GB) for 24h.

4 Results

For the evaluation of our new sequence to sequence approach and the transformer model we built, we use the coreference official scorer (Pradhan et al., 2014) to get the results of the most used metrics on the task on the ARRAU testing split. We obtain 77.2 F1 at mention detection (MD), 64.9 F1 at MUC, 66.5 F1 at B^3, 65.3 F1 at CEAF$_e$ and 65.6 F1 on the CoNLL metric. They are quite good results for a simple approach which does not use any external information as pretrained word embeddings or LMs, or any linguistic, distance or speaker features other than the auxiliary dataset we used, which just added the amount of raw text and its coreferential relations we had. Our model is able to detect most of the mentions, including singletons, and it does cluster correctly correferential mentions to a certain extent, including those that are at a very long distance[2].

	MD	MUC	B^3	CEAF$_m$	CEAF$_e$	BLANC	LEA	CoNLL
This work	77.2	64.9	66.5	66.7	65.3	59.9	58.0	65.6
(Yu et al., 2020)	—	78.2	78.8	—	76.8	—	—	77.9

Table 8: Our F1 results in comparison with previous best results on the ARRAU dataset.

The best results on the ARRAU dataset are those presented at Yu et al. (2020). Results obtained in this work are not completely comparable with our work, as we do not process documents longer than 1024 tokens (∼800 words, keeping 72% of the documents), while they only test their system with the RST subset of the test set. However, we include the comparison in table 8, to put our results into context, and as we can see, we are not able to match their results.

5 Conclusions and Future Work

All in all, in this work we present a novel approach, as far as we know, the first time where coreference resolution has been learned as a simple sequence to sequence task, using just a Transformer, an architecture that rules the NLP field. We got 65.6 F1 CoNLL on the ARRAU corpus, and despite not getting the best results on the dataset, we proved that a Transformer is enough to learn the task, from raw text, without any features or pre-trained word-embeddings or LMs. The results obtained are quite good, as this approach have room for improvements at architecture level, hyperparameter tuning, and the integration of pretrained LMs. This approach may help at unifing the coreference resolution with other NLP models, where this task could be used at pretraining sequence to sequence LMs (T5). Our code and model are available at: `https://github.com/gorka96/text2cor`.

There are many aspects of this approach worth to continue researching. To begin with, we limited the maximum length of the sequences to 1024 tokens for simplicity, nevertheless, to be able to process longer documents, we will need to train Transformer models with longer maximum positions. To handle the increment in memory and computational costs, architectures that do not use full attention as reformer (Kitaev et al., 2020) or longformer (Beltagy et al., 2020) could be considered. Moreover, we would like to verify that this method is as universal as we said here, trying datasets without singletons, low-resourced languages, and multilingual or cross-lingual settings. Finally, using this approach to train a sequence to sequence language model like T5, would be interesting.

Acknowledgements

This research was partially supported by the Department of Industry of the Basque Government (DeepText project, KK-2020/00088) and by the European Commission (LINGUATEC project, EFA227/16). We thank the three anonymous reviewers whose comments and suggestions contributed to improve this work.

[2]Sample of the output: `https://github.com/gorka96/text2cor/blob/main/pred_example.txt`

References

Samarth Agrawal, Aditya Joshi, Joe Cheri Ross, Pushpak Bhattacharyya, and Harshawardhan M Wabgaonkar. 2017. Are word embedding and dialogue act class-based features useful for coreference resolution in dialogue. In *Proceedings of PACLING*.

Vinay Annam, Nikhil Koditala, and Radhika Mamidi. 2019. Anaphora resolution in dialogue systems for south asian languages. *arXiv preprint arXiv:1911.09994*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. Preco: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. 2018. Exploring Spanish corpora for Portuguese coreference resolution. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 290–295. IEEE.

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.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665.

Loïc Grobol. 2019. Neural coreference resolution with limited lexical context and explicit mention detection for oral french. In *Second Workshop on Computational Models of Reference, Anaphora and Coreference*, page 8.

Samira Hourali, Morteza Zahedi, and Mansour Fateh. 2020. Coreference resolution using neural mcdm and fuzzy weighting technique. *International Journal of Computational Intelligence Systems*.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S Weld. 2019. Bert for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5807–5812.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Mateusz Kopeć. 2019. Three-step coreference-based summarizer for polish news texts. *Poznan Studies in Contemporary Linguistics*, 55(2):397–443.

M Hari Krishna, K Rahamathulla, and Ali Akbar. 2017. A feature based approach for sentiment analysis using svm and coreference resolution. In *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 397–399. IEEE.

Gourab Kundu, Avi Sil, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual coreference resolution and its application to entity linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 395–400.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Bartłomiej Nitoń, Paweł Morawiecki, and Maciej Ogrodniczuk. 2018. Deep neural networks for coreference resolution for Polish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, pages 395–400.

Takumi Ohtani, Hidetaka Kamigaito, Masaaki Nagata, and Manabu Okumura. 2019. Context-aware neural machine translation with coreference information. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 45–50.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Sameer Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A Unified Relational Semantic Representation. In *Proceedings of the International Conference on Semantic Computing*, (ICSC '07), pages 517–526, Washington, DC, USA. IEEE Computer Society.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 1–27, Portland, Oregon.

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*, pages 1–40, Jeju Island, Korea, July. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland, June. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Hossein Sahlani, Maryam Hourali, and Behrouz Minaei-Bidgoli. 2020. Coreference resolution using semantic features and fully connected neural network in the persian language. *International Journal of Computational Intelligence Systems*, 13(1):1002–1013.

A Sboev, R Rybka, and A Gryaznov. 2020. Deep neural networks ensemble with word vector representation models to resolve coreference resolution in russian. In *Advanced Technologies in Robotics and Intelligent Systems*, pages 35–44. Springer.

Tomohide Shibata and Sadao Kurohashi. 2018. Entity-centric joint modeling of japanese coreference resolution and predicate argument structure analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 579–589.

Josef Steinberger, Mijail Kabadjov, and Massimo Poesio. 2016. Coreference applications to summarization. In *Anaphora Resolution*, pages 433–456. Springer.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 2019. Deep cross-lingual coreference resolution for less-resourced languages: The case of basque. In *Proceedings of the 2nd Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2019), co-located with NAACL 2019*.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in a variety of genres: the arrau corpus. *Natural Language Engineering*, 26(1):95–128.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Using coreference links to improve spanish-to-english machine translation. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 30–40.

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.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020. A cluster ranking model for full anaphora resolution. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 11–20.

Pengfei Zhu, Zhuosheng Zhang, Jiangtong Li, Yafang Huang, and Hai Zhao. 2018. Lingke: a fine-grained multi-turn chatbot for customer service. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 108–112.