A Deep Architecture for Coreference Resolution

Xiao Cheng
Computer Science Department
xiao@cs.stanford.edu

Rob Voigt
Linguistics Department
robvoigt@stanford.edu

Abstract

The automatic resolution of surface forms into clusters which co-refer to the same abstract entity is a difficult task with a long history in computational linguistics. Existing systems for this task tend to rely on complex linguistically-motivated rules or statistical information about pairs of mentions to make coreference decisions; these systems can be brittle, and tend to have difficulty with high-level abstract semantics, an area where deep models may be expected to be of use. In this work, we present a deep architecture for coreference resolution using LSTMs which requires virtually no hand-engineering of features and potentially better models human processes of reference resolution. We evaluate our system on the standard CoNLL 2012 shared task dataset and compare to existing models. While our results are not state of the art, we surpass some existing published hand-engineered systems and show that this line of research is a promising direction for this difficult task.

1 Introduction

In natural languages, speakers use a wide variety of surface forms for linguistic reference, including diverse forms of a noun, titles, abbreviations, pronominalization, and more. Coreference resolution is the task of automatically resolving all overt referential mentions into clusters, each of which consists of mentions co-referring to the same abstract entity.

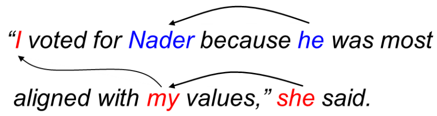
"I voted for Nader because he was most
aligned with my values," she said.

Figure 1: Visualization of the task. Source: Stanford NLP Group

In the example in Figure 1, "Nader" and "he" are members of one cluster, and "I," "my," and "she" are members of another. Due to the diversity of surface forms for linguistic reference and special phenomena such as pronominal coreference, this task is very difficult, and performance of state-of-the-art systems remains far from human levels: below 65% on the standard CoNLL metric, which is an average of three coreference metrics.

Existing systems for this task tend to use either simple surface features in pairwise classification architectures [1, 2, 3] which can lack the necessary expressivity for the complexities of linguistic reference, or they rely on deterministic hand-written rules [4, 5, 6] which are relatively brittle and do not transfer well to new domains.

In this work we propose that coreference is best viewed as a memory-based sequential learning task with a document-level loss function. Such an architecture has the advantage, first of all, that it is analogous to human on-line processing of coreference, where listeners maintain a state of common ground about the entities under discussion, and probabilistically infer the intended reference given the previously seen sequence [7, 8]. Furthermore, learning a deep model for this task would removes the necessity for hand-engineering of complex features traditionally required in coreference resolution.

2 Related Work

Early work in coreference resolution often derived from linguistic research on anaphoricity in natural languages, implementing linguistic theories with complex sets of rules [9]. Perhaps the most famous such early example is Hobbs' algorithm for proniminal resolution, based primarily on deterministic walks through syntactic trees [10]. Other unsupervised systems interpreted nominal coreference resolution as a clustering task, where decisions are made sequentially based on constraints regarding attributes such as head words, gender, animacy, number, definiteness, and so on [11].

The arrival of statistical machine learning changed the field, with learning-based methods beginning to dominate in the late 1990s; the features used were often analogous to those generated by deterministic rules, but placed into a learned classification context, often with an architecture focused on pairwise comparisons between mentions [12, 13]. These approaches often also included additional modules for sub-tasks such as anaphoricity detection – the determination of whether a given mention is likely to be coreferent with anything or not [14].

These purely statistical, supervised methods were somewhat upturned in the early 2010s by the advent of sieve-based architectures, which use multiple passes over each document, applying "sieves" of differing deterministic rules in a fine-to-coarse fashion; that is, merging mentions into clusters first based on highly likely rules, and using these clusters to constrain decisions at later stages when the rules are lower-precision, such as in the case of pronouns [5, 15, 6].

Most recently, the purely statistical approach has again gained favor, as a relatively simple log-linear model using feature templates recaptured state-of-the-art performance on the task [3]. In that work, the authors point out that in the proper architecture, traditional pairwise-style feature templates successfully capture a good deal of the syntactic and discourse variation necessary to model coreference; however, semantics present a stumbling block. For example, because exact string match and head-word string match features are so prominent, their system has a great deal of difficulty handling mentions that use previously unseen head words.

This issue suggests that deep models for coreference are a good place to look. Word vector embeddings derived from co-occurrence in corpora have shown high applicability to semantic tasks such as grounding in images, semantic role labeling, and sentiment analysis [16, 17, 18, 19], so perhaps they will provide the semantic representational capacity to address the difficulty of semantics in coreference.

3 Data

For this task we use the English data from the CoNLL 2012 shared task on multilingual coreference resolution, the standard dataset for this task [20].

One substantial limitation in this setting is the dataset is relatively small: the training set contains only 2802 documents with approximately 1.3 million words total.

Hand-annotation of linguistic data for coreference requires substantial training and time, and is therefore relatively expensive to obtain, so this limitation is somewhat unavoidable. Existing systems often use external data sources like WordNet, gender and animacy lexicons, and named entity annotations [3], but these are potentially not robust to new data. In this work we use no external data sources except pre-trained word vector representations.

The CoNLL shared task data is in a relatively complex task-specific format. The first step of this project, therefore, was to implement a coreference data reading and evaluation framework in Python, that provides an easy interface for training, allowing the use of simple nested `for` loops to access the documents in the corpus and the sentences, parses, and tokens for each. In a future iteration of this work we intend to release this code publicly to encourage deep learning researchers to investigate the coreference task; most existing state-of-the-art systems are written in Java [15, 21, 3], while

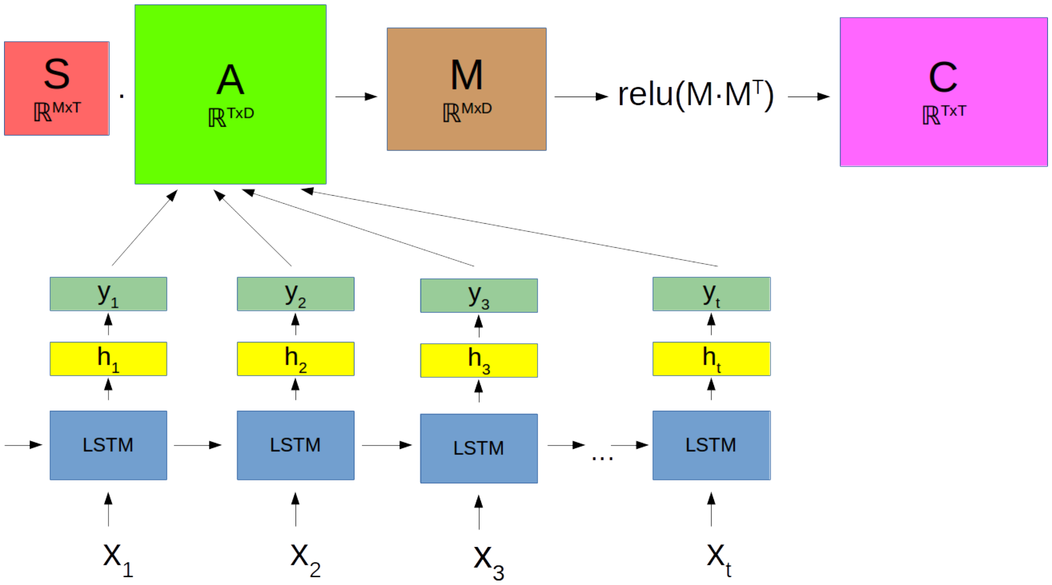

Figure 2: Diagram of the architecture of our system for the generation of the coreference matrix C for one document with T words $(x_1, ..., x_t)$.

Python is much more common for deep systems with packages such as Theano, and the necessity to handle the CoNLL data format may indeed prove a substantial barrier to entry.

4 Architecture

In this project, we propose a new architecture for coreference resolution which combines a word-level sequential memory network with a global loss function that considers pairwise mention-mention links in an entire document.

We base this architecture on two primary intuitions. First, as previously mentioned, since real-world coreference resolution is a sequential task for humans in which we build up mental representations of the entities under discussion and probabilistically link new mentions to existing clusters in this representation, the sequential component of the system has the potential to model coreference analogously to real-world processing. For example, we expect pronouns to act as operators that encourage the model to look into its memory for semantically compatible mentions available for coreference. Secondly, a document-level loss function considering pairwise links may help to encourage global consistency.

To implement such a model we use a long short-term memory (LSTM) setup for sequential modelling [22], where our input is word vectors pretrained with some other method, though these are updated with backpropogation during training in a manner to be described.

To these word vectors we concatenate a few simple additional features to provide some higher-level context to the model. We add binary features that fire if the word is capitalized and if it's the last token in a given sentence, to inform the model about sentence boundaries. We also add index features for the part-of-speech tag of the word and the ID of its speaker in spoken documents.

Given T words in a document, we run the LSTM over all the words in the document to generate a set of hidden state vector representations, which we multiply by another weight matrix W to generate our word-level outputs ($y_1, ..., y_t$ in Figure 2). These we concatenate into into a matrix A in $\mathbb{R}^{T \times D}$, where D is the dimensionality of the hidden state.

We then define another matrix S of dimensionality $\mathbb{R}^{M \times T}$, where M is the number of mentions in the document. S is a matrix that serves to index which words belong to coreferent mentions in the document. We want mentions to be represented by the average of their constituent word-level representations, so a given row in S will contain all zeroes except for n adjacent entries with the value of $\frac{1}{n}$ each. The generation of this matrix is analogous to the "mention detection" step in many existing systems [15, 3].

We take $S \cdot A$ to produce a matrix M in $\mathbb{R}^{M \times D}$, where rows correspond to coreferent mentions in the document. In our current formulation, each mention is represented in M by the average of its constituent word-level output vectors; each row is a mention, maintaining dimensionality D of the size of the hidden state output from the LSTM. We normalize the rows in M to length one.

Then we use M to generate what we term a "coreference matrix," somewhat analogous to a correlation matrix, by taking MM^T and passing the matrix through a rectified linear unit (RELU) pointwise nonlinearity. This produces our coreference matrix C in $\mathbb{R}^{M \times M}$, where we consider the value for a given entry in the matrix $M_{i,j}$ – effectively, the cosine similarity between mentions representations i and j to be the "coreference score" between mentions i and j. The process described above through generation of C is shown in Figure 2.

At training time, we compare C to a gold coreference matrix G, which is composed of zeroes and ones: $M_{i,j}$ is zero when mentions i and j do not occur in the same cluster, is one when they do. We use a loss function with pointwise cross-entropy, expressed as follows:

$$L = -\frac{1}{N^2} \sum_{i,j} T_{i,j} * \log C_{i,j} + (1 - T_{i,j}) * \log(1 - C_{i,j})$$

At decoding time, we set a threshold γ on the entries in C: entries above the threshold are marked as one (or, coreferent) and entries below are marked as zero. We use C to then generate a coreference graph where all one-valued mention pairs are grouped into the same cluster. Single singleton mentions are not annotated in the gold coreference data, any mentions with only entries below the threshold are marked as singletons and not included in the graph. We anticipate that this threshold will directly allow us to choose a tradeoff point on the precision-recall curve; the threshold is tuned empirically on the development data to set the optimum tradeoff.

5 Experiments

In our experiments we compare to results from the CoNLL 2012 shared task; in particular, the closed-track results on English with gold mentions provided. We implement the architecture described above in Theano [23].

We use only the provided training data, and therefore our results are comparable to the closed track. As mentioned above, the generation of the words-to-mentions binary matrix S is analgous to the mention detection step in many systems submitted to the shared task. For the purposes of this project we omit this step, and directly use the gold mention spans provided in the training and testing data.

All systems in the shared task also reported results under this condition, so this allows us to make direct comparisons without introducing unnecessary noise. Mention detection itself may also be amenable to learning by deep models, but we do not include this component in this project.

Evaluation results are reported using the CoNLL coreference evaluation metric. Since coreference outputs are complex graphical structures, evaluation of coreference itself is a difficult question. The CoNLL metric is simply an unweighted average of three separate metrics: MUC, B-CUBED, and CEAF.

In these experiments, we use 50-dimensional word vectors trained with GloVe [24] on 6 billion tokens of English, and use a hidden state D of size 128.

5.1 Results

Our results on the CoNLL 2012 test set are given in Table 1. These results are generated with our threshold γ set to the average best threshold tuned on the development set, which empirically we found to be 0.79.

SYSTEM	MUC			B^3			CEAF$_e$			SCORE
	P	R	F1	P	R	F1	P	R	F1	
Li (2012) [25]	-	-	60.45	-	-	57.18	-	-	36.58	50.65
Zhekova et al. (2012) [26]	68.54	78.10	73.01	63.14	58.63	60.80	52.84	37.44	43.83	59.21
Fernandes et al. (2012) [2]	71.18	91.24	79.97	65.81	85.51	74.38	74.93	43.09	54.72	69.69
Chen and Ng (2012) [27]	72.3	89.4	79.9	64.6	85.9	73.8	76.3	46.4	57.7	70.5
Durrett and Klein (2013) [3]	-	-	84.49	-	-	75.65	-	-	69.89	76.68
This work	64.6	71.08	67.68	59.22	49.12	53.7	41.2	39.46	40.31	53.89

Table 1: Our results compared to performance of existing systems on the CoNLL 2012 shared task data. Results for all systems are given using gold mentions for comparability with our results. - is used when a particular paper did not report a particular result.

6 Conclusion

In the end, our performance was reasonable; our system is comparable to the systems on the lower end of the spectrum for the shared task [25, 26], and definitely a substantial ways away from the state of the art. Nevertheless, our system achieved this reasonable performance with virtually no hand-engineering of features that all the systems we are comparing to contain. We use no attribute labels, no named entity recognition, and almost no external information except pre-trained word vectors.

Furthermore, our system's performance arises in spite of the fact that it does not explicitly encode any information about exact string matches and head word matches, which are by far the strongest features for nominal coreference in traditional systems. One could imagine extending our current system by adding additional explicit features of the type seen in more traditional systems into the mention matrix M to encode some of this linguistic knowledge.

As shown in Figure 3, our system is successfully able to learn coreference links between semantically-related mentions that are abstractly connected but have few overt cues to their reference. For example, in the document given, we link "only about 1,500 in the world" with "a rather special one," a correct coreference link that almost any existing coreference system would be very unlikely to make.

Still, the problem of insufficient training data for coreference systems is a persistent one. While there is fundamentally no way to get around the extreme expense in money and time to produce annotated coreference data, our system provides an entry point for introducing useful external supervision in the form of additional embeddings.

For example, one early possibility that we considered was to use linked textual references in Wikipedia as instances of coreference, and to train a GloVe-like model with a modified loss function encouraging the embedded representations to be more nearby in a "referential" vector space as opposed to the traditional co-occurence space. Since there will likely never be sufficient annotated coreference data to effectively optimize our word-level representations from scratch, in our model such a representation might act as a much better initialization of our parameters and allow the model to find a better local optimum.

We could also reconsider the composition of the mention detection matrix S; at the moment it effectively takes the average of the constituent vectors in A, but one could imagine an instantiation that, for example, explicitly considers the head word and computes a weighted average where the head word is the most important element.

There is a sense in which it seems likely that a more developed system along these lines would use parse tree structures rather than raw words, such as with a TreeLSTM [28]. We considered this possibility, but it remains unclear how to appropriately propagate information from one sentence to the next. Passing the hidden state of the top node of the tree would require the LSTM to remember low-level details such as the presence of a pronoun in a very abstract representation.

This project serves as an effective first proof-of-concept: it is possible that the beneficial semantic properties that emerge from co-occurence-based word embeddings can be leverged effectively with deep learning systems to address the complex task of linguistic coreference. However, we're not there yet, and it may be the case that the most effective long-term solution will involve combining deep models such as the one presented here with traditional, linguistically-informed hand-engineered features and rules. We intend to continue work on this project and produce full, publishable results as soon as possible.

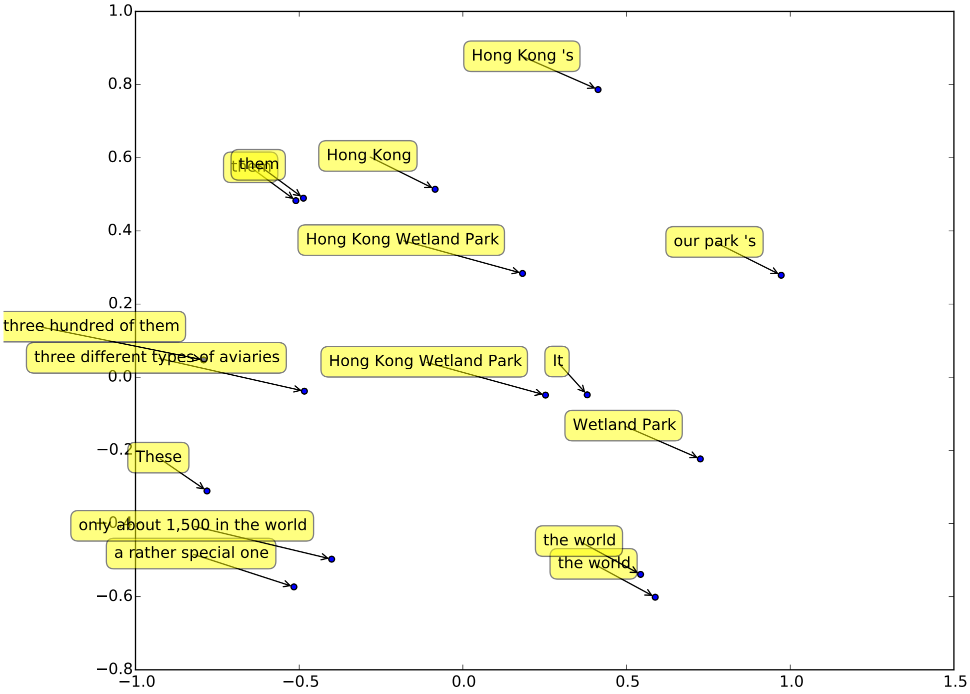

Figure 3: PCA projection of the mention vectors for a document in the development set.

References

[1] Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393. Association for Computational Linguistics, 2010.

[2] Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 41–48. Association for Computational Linguistics, 2012.

[3] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *EMNLP*, pages 1971–1982, 2013.

[4] Sven Hartrumpf. Coreference resolution with syntactico-semantic rules and corpus statistics. In *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7*, page 17. Association for Computational Linguistics, 2001.

[5] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference res-

olution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics, 2010.

[6] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

[7] Joy E Hanna, Michael K Tanenhaus, and John C Trueswell. The effects of common ground and perspective on domains of referential interpretation. *Journal of Memory and Language*, 49(1):43–61, 2003.

[8] Joy E Hanna and Michael K Tanenhaus. Pragmatic effects on reference resolution in a collaborative task: Evidence from eye movements. *Cognitive Science*, 28(1):105–115, 2004.

[9] Shalom Lappin and Michael McCord. Anaphora resolution in slot grammar. *Computational Linguistics*, 16(4):197–212, 1990.

[10] Jerry R Hobbs. Coherence and coreference*. *Cognitive science*, 3(1):67–90, 1979.

[11] Claire Cardie, Kiri Wagstaff, et al. Noun phrase coreference as clustering. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 82–89, 1999.

[12] Joseph F McCarthy and Wendy G Lehnert. Using decision trees for coreference resolution. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, 1995.

[13] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.

[14] Vincent Ng and Claire Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[15] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[18] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[19] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[20] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics, 2012.

[21] Anders Björkelund and Richárd Farkas. Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 49–55. Association for Computational Linguistics, 2012.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU

and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014.

[25] Baoli Li. Learning to model multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 129–135. Association for Computational Linguistics, 2012.

[26] Desislava Zhekova and Sandra Kübler. Ubiu: A robust system for resolving unrestricted coreference. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 112–116. Association for Computational Linguistics, 2011.

[27] Chen Chen and Vincent Ng. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 56–63. Association for Computational Linguistics, 2012.

[28] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.