This paper was submitted as a final project report for CS6424/ECE6424 *Probabilistic Graphical Models and Structured Prediction* in the spring semester of 2016.

The work presented here is done by students on short time constraints, so it may be preliminary or have inconclusive results. I encouraged the students to choose projects that complemented or integrated with their own research, so it is possible the project has continued since this report was submitted. If you are interested in the topic, I encourage you to contact the authors to see the latest developments on these ideas.

Bert Huang
Department of Computer Science
Virginia Tech

ECE 6424 Project: Enforcing Consistent Predictions in Visual Question Answering models

Aroma Mahendru
Faculty Collaborators: Dr. Devi Parikh, Dr. Dhruv Batra
Department of Electrical and Computer Engineering
Virginia Tech

Abstract

The current architectures for the task of visual question answering (VQA) predict reasonable answers for individual questions independent of other questions. However, it is seen that predictions of these models are not consistent over related questions. This is not surprising because there is no explicit mechanism present in these models which ensure consistency across various questions for an image.
The goal of this project is exactly this i.e. devise a mechanism to enforce consistency in the predictions of the VQA models. Specifically, we want to solve this problem: *Given n questions for an image, enforce consistency over the answers of these n questions.* This project proposes a basic approach to solve this problem which uses CRF inference as a post processing step on top of the state-of-the-art VQA model. The hope is that the consistency constraints introduced lead to a more accurate VQA model.

1 Introduction

Visual Question Answering is one of the very challenging tasks which the Artificial Intelligence (AI) community is lately interested in. The idea is that an intelligent machine should be able to answer any question regarding an image which has been shown to it. In a sense, this task is the holy grail of computer vision and natural language processing since it expects complete knowledge about the image from the AI agent and understand natural language questions.
However on a closer look, one can see that this task requires much more than understanding image and the question. For instance, look at the question answer instances in Fig. 1. An AI agent needs to know 20/20 vision has a relation to spectacles (Fig. 1a) and understand that people usually eat pop corn while watching television (Fig. 1b) to answer these questions correctly. Thus, an AI agent also

(a) Q: *Does this person have 20/20 vision?*

(b) Q: *What is this person doing?*

Figure 1: Image/Question instances where an AI agent requires real world common sense for correctly answering the question in addition to solving vision and understanding natural language. (Images from [1, 2])

needs some common sense knowledge about the world and logical abilities. The most basic 'logical ability' is that the model gives consistent predictions for various questions for the same image. For

example, in Fig. 2a, an ideal AI agent should be able to infer that since this is a sandwich, it is a breakfast meal and hence time of the day is morning. Similarly in Fig. 2b, since the girl is playing tennis, she has a tennis racket in her hand.
Introducing human-like logical capabilities in AI agents or models is one of the intuitions for building consistent models. After looking at the examples in Fig. 2, one can easily deduce that the model can make more accurate predictions (and eliminate noise) if it is ensured answers of the related questions are consistent with each other.
Ideally we would want that the model intrinsically learns this capability from the training data. This

(a) Q1: *Is it a breakfast meal?* Q2: *What time of the day is it?*

(b) Q1: *Is she playing tennis?* Q2: *Does she have a racket?*

Figure 2: Image/Question instances with two questions per image. As one can see, these questions are related to each other and an intelligent AI agent would predict *'consistent'* answers for these pairs.(Images from [1])

would obviously require large amount of data. Computer vision and natural language processing communities have come a long way in that aspect. For example ImageNet dataset has 1.2 million training images and VQA dataset has 250k real images and 10 million questions. Unfortunately, it's been seen that even that is not enough for the models to learn these common sense and logic capabilities. Since we don't have enough data, the obvious next step is to encode consistency in our model somehow. This can be done in two ways:

- Explicitly optimize the probability of answers for all the questions asked on the same image with respect to the consistency constraint.
- Redefine training loss function in a way which respects consistency across various questions in an image.

As one can see, the first way applies consistency constraint explicitly while the second way aims to do the same implicitly and so a more ambitious goal. In this project, we have tried to do the former. The idea is to perform inference on a CRF built on top of the VQA model using external knowledge source. Each question (or a set of questions which are same in meaning) is represented as a node in the graph where its states are all the possible answers and its neighbors are other relevant questions. Rest of the paper is organized as follows: Section 2 describes the previous work done related to the tasks of visual question answering, application of CRF inference to various language/vision tasks and finally usage of knowledge bases; the approach and model architecture are described in detail in section 3; experimental setup and results are presented in sections 4 and 5 respectively; the future directions of the work and its implications are discussed in section 6 and finally the conclusions are presented in section 7.

2 Related Work

Visual Question Answering: Due to the growing popularity of multimodal learning, there has been an explosion of many visual question answering datasets such as DAQUAR[3], VQA[1], Visual Madlibs[4], COCO-QA[5], etc. Originally, [3] proposed a method that combines semantic parsing and image segmentation with a Bayesian approach to sample from nearest neighbors in the training set. Researchers[6] have also worked on query answering system based on a joint parse graph from text and videos. Geman et al. [7] proposed an automatic 'query generator' that is trained on annotated images and produces a sequence of binary questions from any given test image. Most recently, inspired by the significant progress achieved using deep neural network models in both computer vision and natural language processing, an architecture which combines a CNN and RNN to learn the mapping from images to sentences has become the dominant trend[8]. Most approaches use RNNs to encode the question and output the answer. In fact, almost all of the top performing

models[9, 10, 1] on these datasets are based on recurrent neural network architectures. Gao et al. [9] used two networks, a separate encoder and decoder, Malinowski et al. [3] used a single network for both encoding and decoding. Ren et al. [10] focused on questions with a single-word answer and formulated the task as a classification problem using an LSTM[8]. People have also used [10] CNNs to both extract image features and sentence features, and fuse the features together with another multimodal CNN. Antol et al.[1] use a CNN+ LSTM too, which encodes the image with CNN features and questions with LSTM representation. Many others [11] encode visual attention in the Image Captioning propose to use the spatial attention to help answering visual questions. Some others[12] take advantage of compositional nature of natural language questions and use deep network 'units' in a modular fashion. However, none of the models yet have tried to incorporate logical consistency to improve the model performance.

CRFs in vision and natural language: Conditional Random Fields (CRFs) are natural choice for many relational problems because they allow both graphically representing dependencies between entities, and including rich observed features of entities. For example , CRFs are heavily used for a variety of natural language processing problems like named entity recognition[13], shallow parsing[14], word alignment in machine translation[15], finding semantic roles in text[16], etc. In computer vision as well, CRFs have been used for the tasks of semantic segmentation[17–19] and object recognition[20]. Thus, CRF was an obvious choice for modeling complex dependencies in the predictions of multiple questions for an image.

Knowledgebases: Recently, a large number of knowledge graphs have been created, including YAGO [21], DBpedia[22], NELL[23], Freebase[24], and the Google Knowledge Graph[25]. Knowledge graphs are already powering multiple "Big Data" applications in a variety of commercial and scientific domains[26]. For example Google's Knowledge Graph, which currently stores 18 billion facts about 570 million entities, into the results of Google's search engine. The Google Knowledge Graph is used to identify and disambiguate entities in text, to enrich search results with semantically structured summaries, and to provide links to related entities in exploratory search. IBM's question answering system Watson, which was able to beat human experts in the game of Jeopardy! used YAGO[21], DBpedia[22], and Freebase[24] as its sources of information amongst others. Repositories of structured knowledge are also an indispensable component of digital assistants such as Siri, Cortana, or Google Now[26]. Knowledge graphs are also used in several specialized domains. For instance, knowledge graphs[27] integrate multiple sources of biomedical information. These have been used for question answering and decision support in the life sciences[26]. It is a common practice in natural language processing research[28, 29] to use knowledge-bases for building language models. Large-scale knowledge bases , such as Freebase[24] and DBpedia [22] have been used successfully in natural language question answering systems. However, usage of knowledge bases in VQA models is still quite rare[8].

3 Approach

Let's start with looking at a standard visual question answering model (See Fig. 3). This model takes in question and image features point-wise multiplied into an LSTM as input and gives output scores/probabilities for a fixed number of most frequent answers. The question features are extracted by mapping the dictionary of all words we can hope to see to a lookup table (i.e. random embeddings) or word2vec embeddings. fc7 features from VGGNet pre-trained on ImageNet dataset are extracted the image. Thus, the whole system i.e. the LSTM is trained in an end-to-end fashion. As we can

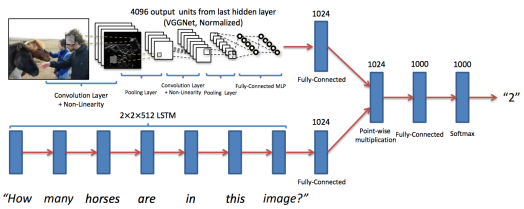

Figure 3: Baseline CNN LSTM visual question answering model. (Figure from [1])

see this model considers every image question pair as an independent query and no two question answer pairs are dependent on each other. However, we already know that question and answer pairs

for a particular image, are a lot of times not independent of another. For those cases, we can build an explicit model to take advantage of this dependency. As explained briefly in section 1, we have modeled these dependencies in a CRF model.

3.1 CRF model

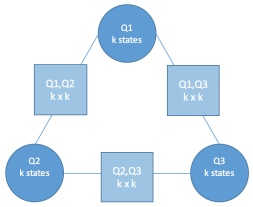

Figure 4: Above figure shows a fully connected CRF for 3 questions with k states.

Consider a visual question answering task, where n *related* questions per image are asked. Our goal is to maximize the joint probability of the predicted answers for all the n questions. Since all the questions are assumed to be related, the most appropriate graph structure is an undirected graph which is fully connected in the pairwise fashion. Thus, there are n nodes and $\binom{n}{2}$ edges in the graph as shown in Fig. 4. The cardinality of each node is the number of possible answers (say k) in the vocabulary of answers of the VQA model. In our case, it's the number of units in the last ($softmax$) layer of the network. Accordingly, the edge potentials would have the size $k \times k$. The scores predicted for individual n answers by the VQA model are used as unary potentials. Edge potentials are learned from an external knowledge base, details are described in subsection 3.2.

3.1.1 Inference

In the graphical models literature, many inference algorithms have been proposed for approximate inference over cyclic graphs. These include sum/max product belief propagation, dual decomposition, alternating directions dual decomposition to mention a few. There are other more sophisticated max margin approaches which have even better guarantees. For this project, since the graph structure is not very complicated, a simple inference algorithm was enough for inferring reasonable marginal probabilities.

3.2 Extracting edge potentials from knowledge corpus

What does it mean to find the $k \times k$ compatibility scores for each between two nodes and their states in the context of question answering? It means to find every entry p_{ij}^{xy} in the edge potential table for nodes x, y having states i, j respectively; where p_{ij}^{xy} means how probable it is for answers i and j to co-occur for the given questions x and y. It is very obvious that such scores should come either from a language model which can predict such scores or from a large knowledge graph which contains real world co-occurrence statistics.

For this, we first use a natural language parser to figure out the structure of the question and the answer. Using this structure, we extract nouns, verbs and filter out the rest of the sentence. This 'tuple extraction' is done for each node state i.e. question-answer pair. After this we need to devise a meaningful query from pairs of these tuples. By meaningful queries, we mean query the words without any repetition which can represent combination of question answer instances well but also small enough to get a decent number of hits on the knowledge base. While experiments with various language corpus, it was found that 2 words or bigrams was the most efficient way to query. Hence, from a set of $t1, t2$ words in tuples for each question-answer instance, we need to find the best bigram query. One could learn a model to do so, however this is a very hard task in itself.

We use a basic assumption to simplify this task. We exploit the fact that most *meaningful* bigrams would have the most number of hits in a knowledge base. Hence, we queried all possible bigrams from the two tuples and took the maximum count as the potential for that state.

4 Experimental Setup

Dataset and Baseline model: To test our model, we used ~11300 questions for ~3700 images from the Visual Question Answering real images validation dataset[1]. There are 3 questions asked for every image. For this specific dataset, we got results on the Multiple Choice answer setting, where the model has to choose one of the 18 choices supplied for each question. We could have also evaluated on the Open Ended answers where each question would then have as many answer states (1000 in the model we used) as one wants. The only difference would be that the cardinalities of the node would be 1000 instead of 18 and the 18×18 potential values that we queried from the knowledge base would be spread across a 1000×1000 sparse matrix. As described before, we used the standard LSTM CNN baseline in [1].

Language tuple extraction: Parsings of natural language questions and answers were generated using Stanford parser [30], which is the state-of-the-art sentence parser. Code to generate bigrams from these parsings was set up in python.

Knowledge base: We used the bigrams from Google Ngrams[31] as the external knowledge source. One can query the corpus to find the number of times a bigram occurred in the Google books corpus. These counts were used as log potentials in the CRF model. Google Ngrams knowledge corpus can be generally queried using their sever online for an order of hundred queries using python and r APIs. However, since the number of queries in our case was much higher ($11300 \times 3 \times 324 \times \#bigrams$), online querying was obviously not possible.
To query offline and very fast, a 'two key' hashtable was built with roughly ~100 million records containing bigrams using PyTables and pyh5 packages. The setting up of this database was very crucial for querying thousands of bigrams very fast. Code and generated datafile will be available for future work.

CRF model and inference: We used ad3[32] which has python wrapper over a C++ library using which one can build arbitrary graph structures and perform inference over it. The inference algorithm used was alternating directions dual decomposition[32] which is one of the standard inference method for graphical models.

5 Results

We performed multiple experiments to modify and study the proposed model. In the first experiment, (See Table 1.), we tried to calibrate unary potentials of the model with the edge potentials by incrementally weighing unary potentials. It can be seen in Table 1, performance improves with higher weights in the unary potentials with peak performance at weight 10^{14}. Here, the model surpasses the baseline by a very modest 0.04%. This means that high signal from the unary potentials is very important to guide the model.

Weight	Accuracy(%)
10^6	1.99
10^8	45.76
10^{10}	46.53
10^{12}	46.53
10^{14}	46.55
Baseline	46.51

Table 1: Accuracy of consistency model when unary potentials are weighed increasingly. Baseline accuracy in the last row.

There are other ways of calibrating unary and edge potentials other than just weighing unary potentials. One way could be using some function of the queried pairwise potentials instead of using raw counts. Accuracies on a couple of functions which we tried are shown in Table 2. It was observed that though these squashing functions did not help much in the accuracy, the requirement of high weights on the unary potentials was reduced. Both $sigmoid$ and $tanh$ functions even showed slight performance improvement over the baseline when unary potentials were weighed more than 10^2.

Function	Accuracy(%)
log	37.76
$sigmoid$	46.50
$tanh$	46.50
Baseline	46.51

Table 2: Accuracy of consistency model when squashing functions are used on queried counts. Baseline accuracy in the last row. Weight of unaries here is 10^2.

By looking at Table 1, it can also be seen that consistency constraints mostly harm the accuracies than improving them, however there are few questions which do get some benefit of the constraints. Fig. 5 shows some qualitative examples for the same.

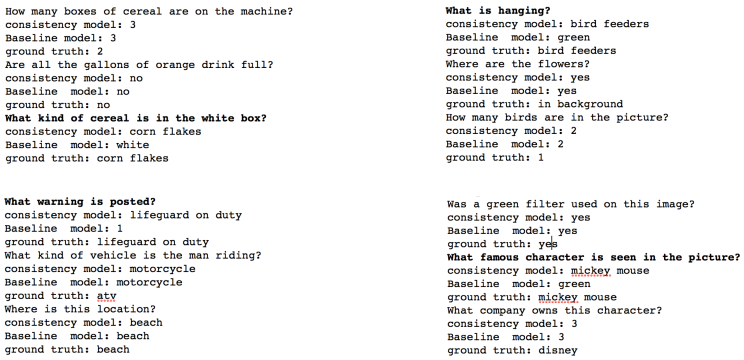

Figure 5: Questions in bold are the questions correctly answered by the consistency model but not the baseline model. The other two questions are question sharing edges with that question.

It can be seen that these questions which were answered correctly didn't change labels because of the adjacent nodes changing labels. But in fact, the reason the labels are flipped is because edge potentials are capturing real world statistics for not just compatibility of states of the nodes but also of the nodes themselves. For example, if all the edge potentials corresponding to the state "corn flakes" are high (which is possible since "cereal, corn flakes" should have a very high count in the knowledge corpus), then the marginals would end up very high for this answer.

6 Discussion and Future Work

There are many observations made during the experiments of this project which will definitely be helpful in the future works in this direction. First of all, as seen from the accuracies and qualitative examples, the consistency model helped in very slight improvement in accuracies. Also the source of this improvement is also not what we were expecting. It was hoped that the information in the answers of the other questions would help in choosing the correct answer for the question. However, what we observe is that whatever improvement we get is by introducing more domain knowledge from the knowledge corpus about the real world data.

However, we made a key assumption here that all the questions collected for an image are related. It turns out that this assumption was not quite true for the dataset we experimented on. Since, there were only three questions per image and there were multiple objects per image; it was hard for questions to be relevant to each other. Unfortunately, most visual question answering datasets have not collected data with the focus of having relevant questions per image. Hence efforts can be made in the direction of data collection. A better solution however would be to train a language model which could take a question (or a sentence) as input and output all the implied questions (or sentences). Such a model could work as a question generator which gives out relevant questions.

Some modifications can be made to the current model as well. For example, instead of querying all possible bigrams for question answer pairs, one could find the most useful pair by checking how close the words in the word2vec or some other similar embedding space. The idea would be that if the words are close in this embedding space, they should form a meaningful bigram to query. Other

than this, one could also use the counts obtained for all possible bigrams as features and train a model which outputs single score for the question answer pair. This model is thus a model which learns edge potentials from counts queried from the knowledge base. An advantage of this model would be that it could integrated in an end-to-end pipeline if need be.

Finally, one could also integrate the CRF as an LSTM in the deep model. Formulation of CRF as RNN has recently been done in [33] where each iteration of mean field algorithm has been interpreted as a stack of CNN filters. This technique was however applied on semantic segmentation is somewhat different from our task. This would be another interesting future direction.

7 Conclusion

In this project we tried to enforce consistency over multiple predictions of a visual question answering model in hope of introducing logic-like capabilities in the model and also improve upon the performance. This was done by constructing a fully connected CRF model whose edge potentials were extracted from an external knowledge base and unary potentials came from a standard VQA model. It was observed that to obtain decent accuracies, either the unary potentials have to weighed very highly or some squashing function has to be applied on the edge potentials. Very slight improvement in performance was observed which after looking at the qualitative examples was found due to the edge potentials encoding real world statistics and not from the answer of the other connected node.

These observations pointed out some limitations to the dataset, a few modifications to the model and numerous interesting problems for future work in this direction.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *International Conference on Computer Vision (ICCV)*, 2015.

[2] H. Pirsiavash, C. Vondrick, and A. Torralba, "Inferring the why in images," *CoRR*, vol. abs/1406.5472, 2014.

[3] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), pp. 1682–1690, Curran Associates, Inc., 2014.

[4] L. Yu, E. Park, A. C. Berg, and T. L. Berg, "Visual Madlibs: Fill in the blank Image Generation and Question Answering," *arXiv preprint arXiv:1506.00278*, 2015.

[5] M. Ren, R. Kiros, and R. S. Zemel, "Exploring models and data for image question answering," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2953–2961, 2015.

[6] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S. C. Zhu, "Joint video and text parsing for understanding events and answering queries," *CoRR*, vol. abs/1308.6628, 2013.

[7] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," vol. 112, no. 12, pp. 3618–3623, 2015.

[8] Q. Wu, P. Wang, C. Shen, A. van den Hengel, and A. R. Dick, "Ask me anything: Free-form visual question answering based on knowledge from external sources," *CoRR*, vol. abs/1511.06973, 2015.

[9] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? dataset and methods for multilingual image question answering," *CoRR*, vol. abs/1505.05612, 2015.

[10] M. Ren, R. Kiros, and R. S. Zemel, "Image question answering: A visual semantic embedding model and a new dataset," *CoRR*, vol. abs/1505.02074, 2015.

[11] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," *CoRR*, vol. abs/1511.02274, 2015.

[12] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Deep compositional question answering with neural module networks," *CoRR*, vol. abs/1511.02799, 2015.

[13] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, (Stroudsburg, PA, USA), pp. 188–191, Association for Computational Linguistics, 2003.

[14] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, (Stroudsburg, PA, USA), pp. 134–141, Association for Computational Linguistics, 2003.

[15] P. Blunsom and T. Cohn, "Discriminative word alignment with conditional random fields," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, (Stroudsburg, PA, USA), pp. 65–72, Association for Computational Linguistics, 2006.

[16] D. Roth and W.-t. Yih, "Integer linear programming inference for conditional random fields," in *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, (New York, NY, USA), pp. 736–743, ACM, 2005.

[17] S. X. Yu, R. Gross, and J. Shi, "Concurrent object recognition and segmentation by graph partitioning.," in *NIPS*, 2002.

[18] E. Borenstein and S. Ullman, "Combined top-down/bottom-up segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[19] C. Wojek and B. Schiele, "A dynamic conditional random field model for joint labeling of object and scene classes," in *European Conference on Computer Vision (ECCV)*, 2008.

[20] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *In NIPS*, pp. 1097–1104, MIT Press, 2004.

[21] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, (New York, NY, USA), pp. 697–706, ACM, 2007.

[22] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, (Berlin, Heidelberg), pp. 722–735, Springer-Verlag, 2007.

[23] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning.," in *AAAI* (M. Fox and D. Poole, eds.), AAAI Press, 2010.

[24] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, (New York, NY, USA), pp. 1247–1250, ACM, 2008.

[25] A. Singhal, "Introducing the knowledge graph: things, not strings," *Official Google Blog, May*, 2012.

[26] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *arXiv preprint*, vol. arXiv:1503.00759, 2015.

[27] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette, "Bio2rdf: Towards a mashup to build bioinformatics knowledge systems," *J. of Biomedical Informatics*, vol. 41, pp. 706–716, Oct. 2008.

[28] K. Bellare and A. Mccallum, "Learning Extractors from Unlabeled Text using Relevant Databases," in *Sixth International Workshop on Information Integration on the Web (IIWeb)*, 2007.

[29] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

[30] M.-C. de Marneffe and C. D. Manning, "The stanford typed dependencies representation," in *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, (Stroudsburg, PA, USA), pp. 1–8, Association for Computational Linguistics, 2008.

[31] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov, "Syntactic annotations for the google books ngram corpus," in *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, (Stroudsburg, PA, USA), pp. 169–174, Association for Computational Linguistics, 2012.

[32] A. Martins, M. A. T. Figueiredo, P. Aguiar, N. A. Smith, and E. P. Xing, "Ad3: Alternating directions dual decomposition for map inference in graphical models," *Journal of Machine Learning Research*, vol. 16, pp. 495–545, March 2015.

[33] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," *CoRR*, 2015.