Information Extraction Phase 1 Write-Up

Jacob Johnson

1 Information Extraction Task:

My system performs a relation extraction task to recognize Drug-Drug interactions in literature from the biomedical domain of Pharmacovigilance. (Herrera-Zazo et al) My system takes as input raw documents (eithin within the command line or from a file) and optionally, the set of gold entities within the document (If gold entities are not provided, then my system uses a trained spaCy NER model to extract entities). My system then outputs the set of extracted relation tuples, along with which type of relation it is labeled as and which sentence within the document the relation was extracted from. In accordance with the corpus labeling schemes of Herrera-Zazo et al, there are four labels for relations: "mechanism", which describes how drugs interact at a biological level, "effect", which describes what the interaction does to the patient, "advise", which is just a written encouragement not to allow the drugs the chance to interact, or "int", short for interaction, which simply describes that an interaction is observed between the drugs, without further detail.

2 Resources List:

All external python packages were already available on the CADE lab machines and can be installed using Python's pip package installer, but the urls to the documentation websites are included here.

1. spaCy (`https://spacy.io`): used for dependency parsing, lemmatizing, and vectorization. I also trained its NER capabilities to recognize Drug Entities in the case where gold Entities are not input with the document, where it achieved an F-score of .797.
2. numpy (`https://numpy.org`): used to hold vectors produced by spaCy with the ndarray class, also used to find standard deviations with the numpy.std() function.
3. scipy (`https://scipy.org`): used for calculating cosine similarity between vectors with the function scipy.spatial.distance.cosine().
4. From the python standard library, the pickle package was used to save and load machine learning models, pre-calculated vectors, and forms of the training, dev, and test data that were preloaded from their XML format, the statistics package was used to calculate mean values, the xml package was used to parse the corpus annotations, and the re package was used for regular expressions.
5. The corpus was no longer available at the link provided in Herrera-Zazo et al, but I was able to download it from `https://github.com/isegura/DDICorpus`

3 Technical Description:

My system relies on a vector-semantics evaluation of the syntactic context (dependency path) that connects pairs of entities within a sentence. Lemmas of syntactically-relevant words were recorded. For example, in the sentence, ”A daily dose of 2 mg of coumaphos/kg of body weight for 6 days did not affect the plasma enzymes or the antiprothrombinemic effect of bishydroxy-coumarin in wethers.”, the dependency path extracted by spaCy between ”coumaphos” and ”bishydroxy-coumarin” is kg-of-dose/affect/of-effect-enzyme, where kg-of-dose is the dependency path from ”coumaphos” to ”affect”, of-effect-enzyme is the dependency path from ”bishydroxy-coumarin” to ”affect”, and ”affect” is the most recent common ancestor in the dependency tree of the two entities (also, in this case, the root of the sentence). It is worth noting that not all of these parses are entirely desirable, other research into medical domain relation extraction have expressed concern over the parseability of dense medical data (Quirk & Poon), and a future more syntactically aware implementation of this path extraction could take the ”not” into account, as well as ignoring the inclusion of ”enzyme” in the ”bishydroxy-coumarin” path.

Vectors were computed for context path using spaCy's vectorization, and each of the following three vector aggregation functions:

1. Uniform-Weight: The vectors of all elements of the path were simply added together.

2. Peak-Weight: The vector of the highest element in the dependency tree (”affect” in our example) was given full weight, but weights were attenuated for path elements lower in the dependency tree, as a fraction of the number of elements on that side of the path. For example, ”kg”, ”of”, and ”dose”, as well as ”of”, ”effect”, and ”enzyme” each had a weight of $\frac{1}{4}$, $\frac{2}{4}$, and $\frac{3}{4}$, respectively. If ”kg” had not been included in the dependency path, ”of” and ”dose” would instead have had a weight of $\frac{1}{3}$, and $\frac{2}{3}$, respectively. The intuition here was that possibly the word that connected the entities together syntactically would have the most important bearing on whether a syntactic context was indicative of a relation.

3. End-Weight: The weights of the vectors along the sides of the path are reversed from the ”Peak-Weight” setup, giving the elements of the path nearest the entities the highest weight. For example, ”of”, ”effect”, and ”enzyme” have weights of $\frac{3}{4}$, $\frac{2}{4}$, and $\frac{1}{4}$ respectively. The peak element of the path has the lowest weight at 1/(the multiplied lengths of both sides of the paths), so in this example, ”affect” has a weight of $\frac{1}{9}$. The intuition here was that possibly the words that attached the dependency path to the entities would have the most important bearing on whether a syntactic context was indicative of a relation.

Ultimately, little to no difference was found using the different vector weighting schemes, even when a bagged system incorporating all three schemes was tested.

My system compares the context vector of a given pair of entities against all known gold context vectors in each category. It assigns the label of the group of gold vectors with the highest mean similarity to the new vector, provided the distributions of similarities also fulfill the following two requirements:

1. The mean cosine similarity of the chosen label's gold exemplar context vectors to that of the entities must exceed some threshold. Experiments on the devset suggested an optimal threshold of .55. The intuition here is that if none of the gold context vectors are particularly similar to the entities' context vector, the entities are probably not in a relation.

2. Two ANOVA tests were performed on the similarity scores of the four categories, one including and one excluding the category with the highest mean. If excluding the selected category resulted in a lower F-statistic than including it, it meant there was higher evidence to conclude that the three non-maximal categories had the same mean than there was to conclude that all four had the same mean, suggesting that the mean of the maximal category was actually above the mean of the other three categories in a statistically measurable way. The intuition behind this is similar to the intuition behind mutiple category systems (Thelen & Riloff): if a relation appears to be about equally similar to all categories, it is likely not a member of any of those categories. Experiments on the devset demonstrated minimal performance improvements, however.

4 Evaluation:

For the evaluation, the threshold for minimum highest mean was set to .55, Peak-Weighted vector aggregations were used, and the ANOVA filter was applied. Precision, Recall, and F-score were calculated on a sentence-by-sentence basis, and on a cumulative basis across all gold relations in the test set, but because there were very few repeated relations in the test set, these were minimally different, so only the sentence-level aggregated statistics are reported below. The scores are reported with input gold entities and without, i.e., using spaCy's NER:

	mechanism			effect			advise			int			total		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
spaCy:	.235	.352	.282	.297	.217	.251	.212	.564	.308	.236	.481	.316	.237	.361	.286
gold:	.255	.359	.298	.345	.212	.262	.218	.564	.315	.257	.519	.344	.255	.365	.301

As expected, the relation extraction system works better when it has the gold entities to work from. Interestingly, Recall in the advise and int category of relations were much higher than in all other categories, this is likely due to there being fewer relations in those two categories than in the other two categories in the corpus overall.

5 References:

Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., & Declerck, T. (2013). The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. Journal of biomedical informatics, 46(5), 914-920.

Quirk, C., & Poon, H. (2016). Distant supervision for relation extraction beyond the sentence boundary. arXiv preprint arXiv:1609.04873.

Thelen, M., & Riloff, E. (2002, July). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002) (pp. 214-221).