CIS 530 Project - Identifying Words to Anonymize in Spanish Medical Corpus
MILESTONE 3

Gaurav Kumar, Bhavna Saluja, Ipek Ilayda Onur, Srinivas Suri

Literature Review

1. Kaung Khin, Philipp Burckhardt, Rema Padman, "A Deep Learning Architecture for De-identification of Patient Notes: Implementation and Evaluation", arXiv:1810.01570 [cs.CL], 3 Oct 2018, 15 pages

In this paper, the authors have presented a deep learning architecture that uses bi-directional long short-term memory networks (Bi-LSTMs) with variational dropouts and deep contextualized word embeddings while also using components such as traditional word embeddings (Glove),
character LSTM embeddings and conditional random fields. Their architecture can be broken down into four distinct layers: pre-processing, embeddings, Bi-LSTM and CRF classifier. In the pre-processing layer, they break down the input document into sentences, tokens, and characters. They then use NLTK to generate a part-of-speech (POS) tag for each token. In embedding layer, for each token they concatenate the GloVe (300 dimensions) and ELMo (1024 dimensions) representations to produce a single 1324-dimensional word input vector. The concatenated word vector is then further concatenated with the character embedding vector (50 dimensions), POS one-hot encoded vector (20 dimensions), and the casing embedded vector to produce single 1394-dimensional input vector that is fed to Bi-LSTM layer. Their Bi-LSTM layer consists of 100 hidden units, uses variational dropout and have a dropout probability of 0.5. Their approach in CRF layer is identical to the Bi-LSTM-CRF model by Huang et al (*Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," CoRR, 2015*).

The two main data sets that they used to evaluate their architecture are the 2014 i2b2 de-identification challenge data set and the nursing notes corpus. On the task of binary F1 scores, their architecture performs similarly to the best scores achieved by other architectures with them having a slight edge on precision metrics. They have achieved precision of 0.9830, recall of 0.9737, and binary F1 score of 0.9783. For the nursing corpora, they managed to best the scores achieved by the deidentify system while also achieving a binary F1 score of over 0.812. For majority of HIPAA-PHI categories, their system performed better than system by Yang et al.

2. Amund Tveit, Ole Edsberg, Thomas Brox Røst, Arild Faxvaag, Øystein Nytrø, Torbjørn Nordgård, Martin Thorsen Ranang and Anders Grimsmo, "Anonymization of General Practioner Medical Records", HelsIT, 5 pages

The paper aims to develop techniques and methods for semi-automated anonymization of medical record information. Some anonymization challenges, including linguistic issues (e.g. spelling and ambiguity) and determining which parts of the data that is sensitive are also discussed. The paper proposes methods like utilization of database structure, dictionaries, heuristics and natural language processing for anonymizing patient records in general. The dataset is in Norwegian language which means the task is even more challenging since it is different than English in terms of linguistic aspects. Major challenges which are posited are the differences in identity markers (e.g. Dr. and Mrs.) and hyphenation patterns in Norwegian, unstructured text, no strictly enforced guidelines for how the data should be encoded.

The goal of this paper is to anonymize general practioner data - both structural and free-text. A 6-step anonymization approach is being proposed-

A. Dictionaries - Words and numbers found in structural data can be extracted into local dictionaries with corresponding type (e.g. of patient names, social security numbers, postal codes, health institution names, health personnel names and locations).

B. Exact Match and Tag - In this step, the focus is on processing the unigram created from the free-text notes. Based on all the local and external dictionaries, a combined dictionary is created where one can look up words and get their corresponding type(s). In order to tag non-textual symbols such, e.g. dates, phone numbers and social security numbers, regular expressions are used.

C. Approximate Match and Tag - Patient records may contain erroneously spelled words, and in many cases they might be only slightly incorrectly spelled. Levenstein distance is used; by going through untagged words in the unigram and allowing an edit distance of 1, candidate misspellings can be found by relating them to the combined dictionary.

D1. Resolve Tagged Words - The words in the unigram should be replaced only if it is an identifying name (e.g. person name). When words have multiple type tags, they have to be replaced or removed if one or more of the tags are of the identifying type.

D2. Handle Untagged Words - Untagged words in the unigram needs to be investigated manually by the local clinician for tagging, the result of this investigation could be additional entries for the local or external dictionaries.

E. Final Result - The final result contains patient record text with identifying entities replaced by pseudonyms. In order to ensure an acceptable level of anonymization the result must be validated according to the requirements that motivated the anonymization in the first place.

3. György Szarvas, Richárd Farkas, Róbert Busa-fekete, "State-of-the-art Anonymization of Medical Records Using an Iterative Machine Learning Framework", Journal of the American Medical Informatics Association Volume 14 Number 5 Sept / Oct 2007, 7 pages

Goal of the paper:

The authors have developed a de-identification model that can successfully remove personal health information (PHI) from discharge records to make them conform to the guidelines of the HIPAA.

Feature set Built:

Authors have used feature set of 5 different categories described below:

1. **Word level features:** Word level features like Word Capitalization, punctuation marks, digits, Roman/Arabic numbers and other common word level feature of phone numbers, fax, age and ID's etc
2. **Frequency Information:** Gathered frequency of tokens from the Internet. Used features like frequency of the token, the ratio of the token's capitalized and lowercase occurrences, the ratio of capitalized and sentence beginning frequencies of the token.
3. **Offline Dictionaries:** 5 different dictionaries collected from the Internet.
4. **Contextual Information:** Sentence position, closest Heading to the section and several other features.
5. **Phrasal Information:** a forecasted class of several preceding words and common suffixes etc.

Classifiers Trained:

Authors have trained 3 different classifiers that used 3 different contextual features and used a voting based mechanism to decide if the word belonged to N.E.R. If any 2 classifiers have predicted the same label, the word is assigned that tag otherwise it's not considered NER.

Authors have used an iterative approach in the following way:

1. In the first iteration, they have collected several named entities from the headings and used these entities as an extra dictionary. They train the classifiers and collected all the texts tagged as entities from Step 1.
2. They used all the texts tagged as entities and added this dictionary set as an extra dictionary for step 2.

Metrics Used:

Authors have achieved a F1-Score of 99%. As our dataset is not the same as their dataset, It's difficult to compare our results with their results. However, we are

excited to consider the feature set that they have used for their implementation and extend it for our dataset.

4. Xiao-Bai Li and Jialun Qin, "Anonymizing and Sharing Medical Text Records", Inf Syst Res. Author manuscript; available in PMC 2018 March 19, 47 pages

There are three categories in medical texts: (i)explicit identifiers (EID) which includes patient name, phone number, and social security number, which can be used to directly identify an individual (ii)quasi-identifier (QID) which includes date of birth, admission/discharge date, hospital, and zip code, (iii)Health and medical details (HMDs) such as symptoms, test results, disease, and medications.

To extract these categories, a three step approach will be used.
Step 1: The feature extractor: It will split the document into terms. Each term will include local features such as term length, part-of-speech, etc; global features such as the position of the term in the document. (e.g., header, body text, heading, etc.); external features regarding such as whether the term belongs to a proper noun list, belongs to a medical concept lexicon, etc

Step 2: Base classifiers: Use multiple classifiers independent of each other from the features that are extracted. Goal of the classifiers are to match the terms to EID, QID, HMD, or irrelevant categories

Step 3: Combine the results of independent classifiers to get final tags of the words

5. Yikun Guo, Robert Gaizauskas, Ian Roberts, George Demetriou, Mark Hepple, "Identifying Personal Health Information Using Support Vector Machines", 2006, 5 pages

Overview:
The authors have developed a SVM model that can successfully remove personal health information (PHI) from discharge records to make them conform to the guidelines of the HIPAA.

Features set description:
The authors have experimented using SVM's to recognise NER data. Authors have built feature set using various features and trained svm on them. They have used token level features like orthographic features, length, pos, kind etc. Additionally, they have used features like date, id, phone number etc. They have used ANNIE Web API to identify

hospitals, people or locations etc. The dataset of this paper's are in english and our dataset is in Spanish. Therefore, we cannot use ANNIE directly as the language is different. However, we have implemented phone number feature. We are tokenizing date as is a number column and there are no ID's in our dataset. Therefore, we could not copy the features as it from the paper, however we tried to reciprocate the features as much as we could trying to mimic the paper.

Classifiers Trained:
Authors have trained their dataset on SVM. However, they have not mentioned about their default parameters. SVM kernel runs in O(n^2) where n is the number of rows because of the kernel matrix computation. We used LinearSVC() which scales in linear time. This provides with several advantages over the linear SVM. First of all, it provides us an opportunity to scale the feature set for future milestones. Second, we haven't deviated from using the svm as linear svc is very similar to linear svm without computing the kernel matrix and it runs in linear time optimization.

Metrics Used:
Authors have achieved a F1-Score of 96%. However, since their dataset is different, it's difficult to compare their F-Scores with ours.

Baseline Paper

We have chosen paper 5 for implementation. There are several reasons why we went ahead with paper 5:

1. From an NLP perspective, this paper forms a very good baseline paper to understand how researchers approach the problem of NER for PHI. As the feature set isn't exhaustive like some other papers, where they had almost 140 features! This paper has the practicality to be implemented for this project.

2. They have achieved an f-score of 96%. Although their data was in english, their implementation looks promising because of their results.

3. They have also used non machine learning features like ANNIE, JAPE grammar rules etc. which broaden our scope of implementation and experimentation. While implementing these features for Spanish language doesn't make sense, this paper forms our inspiration and provided us with a roadmap of using Spanish name dictionaries, Spanish location dictionaries for future extensions.

4. There were few papers which didn't implement any machine learning at all. This paper used SVM for their implementation. This provides us with a way to extend our MS3

implementation for the next milestone by implementing more features or other type of classifiers like Neural Nets/LSTMs etc.

Github Repository Link:

https://github.com/ionur/Cis-530-final-project.git

After cloning https://github.com/ionur/Cis-530-final-project.git, please change the working directory to 'submissionM3' (https://github.com/ionur/Cis-530-final-project/tree/master/submissionM3) and follow the steps mentioned in README.pdf.

Also, please refer to **baseline.pdf** where we included our detailed approach for M3.

Link for the final presentation :

https://docs.google.com/presentation/d/1fWq5xAv3SrCDAZkWzhFfLejaNQbNutaWzkpx2fmtTmY/edit#slide=id.g58186f86f5_1_3

RUNNING BASELINE MODEL

We are preprocessing the data, basically tokenizing the raw medical reports files into sentences and further tokenizing the sentences into words. So, ultimately the raw medical report files are converted into list of list of words with their corresponding NER type tags (BIO format). There are 29 tags/labels for the task which we mentioned in Milestone 2.

Step 1. How to preprocess the raw text files -

Training files -
$python preprocessing_data.py --dataDir ./train/gold/ --train

After this pre-processing step, a pickle file - train_word_ner_startidx_dict.pickle - is created which has the data in the NER type tag format (BIO) which is used for building the model.

Dev files -
$python preprocessing_data.py --dataDir ./dev/gold/ --dev

After this pre-processing step, a pickle file - dev_word_ner_startidx_dict.pickle - is created which has the data in the NER type tag format (BIO) which is used for building the model.

Test files -
$python preprocessing_data.py --dataDir ./test/gold/ --test

After pre-processing step, a pickle file - test_word_ner_startidx_dict.pickle - is created which has the data in the NER type tag format (BIO) which is used for building the model.

Step 2. How to run baseline model (baseline.py)

baseline.py assumes that the data should be in the current working directory in the same folder hierarchy as is submitted with this Milestone 3.

$python baseline.py

How to evaluate -

Evaluation on Training Data -

$ python evaluate.py brat ner ./train/gold/ ./train/system

Report (SYSTEM: system):

SubTrack 1 [NER]	Measure	Micro
Total (401 docs)	Precision	0.9792
	Recall	0.9408
	F1	0.9597

Evaluation on Dev Data -

$ python evaluate.py brat ner ./dev/gold/ ./dev/system

Report (SYSTEM: system):

SubTrack 1 [NER]	Measure	Micro
Total (193 docs)	Precision	0.8868

	Recall	0.8199
	F1	0.852

Evaluation on Test Data -

$ python evaluate.py brat ner ./test/gold/ ./test/system

Report (SYSTEM: system):

SubTrack 1 [NER]	Measure	Micro
Total (156 docs)	Precision	0.8918
	Recall	0.8337
	F1	0.8618
