Advances In Natural Language Processing:
Capsule Networks for Part of Speech Tagging

By
Faiz Rafique

A Thesis Submitted to The Honors College
In Partial Fulfillment of the Bachelors of Science Degree
With Honors in
Computer Science
THE UNIVERSITY OF ARIZONA
M A Y  2020

Approved by:

Dr. Mihai Surdeanu
Department of Computer Science
Acknowledgments

I would like to thank the researchers at the University of Arizona’s CLU lab for their immense help with this project. Mithun Paul and Sandeep Suntwal provided thoughtful and helpful guidance with exploring topics for my research and I learned a lot from their supervision. Additionally I’d like to thank Andrew Lee Zupon for his immense help and guidance on this Capsule Network research such as adapting the software / data to work with our task and overall helping to apply Capsule Networks for Part of Speech tagging. Lastly I want to thank my faculty fellow Dr. Mihai Surdeanu for his superb supervision and thoughtful guidance throughout the year. It was an honor to work under Dr. Surdeanu and his help gave me a deeper appreciation for the work that goes on within Natural Language Processing. Overall, this research has given me beneficial knowledge and skills within the field of Computer Science that I hope to take with me as I start my career outside of school.
Abstract

Part of Speech (POS) tagging tasks have constantly evolved throughout the years within the corpus of linguistics. Many shortcomings and complications have arrived with implementations of POS tagging through various computer algorithms. This task fosters such complications due to the many nuances of understanding language such as words having several different meanings, or maybe a phrase expressing totally different thoughts due to a one word difference. Its factors like these that computational linguist try to incorporate when teaching language to a machine and one common way of conducting this task revolves around utilizations of neural networks. A neural network is essentially a single algorithm made up of many artificial neurons that attempts to mimic the biological neural networks that make up human brain such as certain neurons firing when someone distinguishes a verb from a noun. In this work, we explore the applications of a new type of neural network called a Capsule Network and how its implementation serves as an efficient and promising method of conducting several Natural Language Processing (NLP) tasks. In particular we will explore this neural network architecture as well as its performance on a simple Part of Speech tagging task to show its potential within future works.
# Contents

1 Statement of Purpose and Relevance 5

2 Capsule Network Introduction 6
   2.1 Computer Graphics ........................................ 7
   2.2 Capsules .................................................... 8
   2.3 Hierarchy of Parts ......................................... 9
   2.4 Routing By Agreement ..................................... 11
   2.5 Applications and Other Capsule Network Details .......... 13

3 Part of Speech Tagging 14

4 Discussion 16
   4.1 Related Work Search ....................................... 16
   4.2 Data / Adapting Software .................................. 17
   4.3 Results ...................................................... 18
   4.4 Low Resource Scenarios ................................... 19
   4.5 Conclusion and Future Work ............................... 20
1 Statement of Purpose and Relevance

The purpose of exploring Capsule Networks for Part of Speech tagging centers around furthering the capabilities and results of many Natural Language Processing tasks. By itself POS tagging isn’t the solution to any particular NLP problem however it is “something that is done as a pre-requisite to simplify a lot of [other] different problems” therefore furthering advancements in POS tagging leads to advancements in other tasks (Malhotra, Sachin, and Divya 2018). These other tasks can include applications such as text to speech conversion, or word-sense disambiguation. Beyond this however once we can show the effectiveness of capsule networks within POS tagging many other domains of NLP and even beyond could have advancements with the application of such a neural network.

In terms of NLP neural networks constantly help us with several different problems, many of which people utilize within their day to day lives. Such implications involve advancements of text classification, named entity recognition, semantic parsing, question answering, spell checking, etc. Endless possibilities present themselves and further advancements to the quality of life can persist through improvements within these fields. An application of this research can center around advancing fake news detection. Many political conflicts and discussions have occurred through the dissemination of fake news through the internet. Certain articles containing this fake news can have powerful repercussions in setting such as elections and can negatively impact the livelihoods of individuals in political scenes. Therefore, detecting this fake news through advancements in neural networks can greatly benefit society and serve as a means of lessening its adverse effects. Researchers combat these problems by improving the underlying algorithms utilized to create a neural network so that the machine can better understand the properties of a language.
Overall, this research is relevant as we offer a new and possibly more efficient kind of neural network called a Capsule Network that can help to create better solutions for certain NLP problems. A Capsule Network in our case utilizes several layers of capsules in order to detect a Part of Speech. By using this method we can retain much more information about a word and better capture the many nuances in modern language. Even with these improvements that we will show this work is only the beginning in a wide array of instances that researchers utilize neural networks. In the end Capsule Networks are a fairly new advancement and so any research applying this technology helps researchers to find out its overall capabilities and further improvements within the future.

2 Capsule Network Introduction

In October of 2017 Geoffrey Hinton, a well-renowned English Canadian cognitive psychologist and computer scientist known for his work in the fields of artificial neural networks, released a paper called *Dynamic Routing Between Capsules* with the help of his colleagues (Sabour 2017). This paper introduced a new type of neural network called a Capsule Network which demonstrated considerably better results than a past neural architecture called a Convolutional Neural Network on the MNIST data set. Thus showing the promising capabilities of this new Capsule Network architecture and how its algorithms might expand to fields other than computer graphics.

![Figure 1: A CapsNet architecture for MNIST (Sabour 2017).](image)
2.1 Computer Graphics

In order to understand what Capsule Networks do lets first consider the how computer graphics work. In computer graphics we might have two objects such as a rectangle and triangle that we want to place on a canvas. We tell the computer where on the canvas to draw a triangle and rectangle by giving it a specified x and y position (let’s say x=20 and y=30) and angle of rotation (angle=16) to which a rendering function can take and translate into an image like so:

![Figure 2 Computer graphics visualization (Géron 2017).](image)

Now let’s consider the opposite process called inverse graphics. Within the domain of inverse graphics we do the exact opposite. We give the computer an image with a triangle and a rectangle on it and the computer scans the image to tell us where the rectangle and triangle are (x and y position and angle of rotation).

![Figure 3: Inverse graphics visualization (Géron 2017).](image)
Overall a capsule network is essentially a neural network that tries to perform this concept of inverse rendering. This is important because it generates a symbolic representation of the objects in an image overall providing an interpretable step for image recognition. For example we could give a computer the image in Figure 4 and it will tell us that there’s a triangle and rectangle within it.

### 2.2 Capsules

The network is composed of many capsules, “a capsule is any function that tries to predict the presence and instantiation parameters of a particular object at a given location” (Géron 2017).

![Figure 4: Many capsules predicting the presence of a triangle and rectangle (Géron 2017).](image)

In this picture the blue vectors represent triangle capsules and the black vectors represent the rectangle capsules. The bigger the vector the more likely the desired object was found at a given location as the picture shows with the longer blue and black vectors. In order to implement this first part we apply convolution to the image which essentially scans pieces of the image to figure out what objects we are looking for. Additionally, we want to make sure that these vector’s length is below 1 as the vectors represent an estimated probability of an objects existence and so in order to do this we utilize this squash function.
where $v_j$ corresponds to vector output of a capsule and $j$ and $s_j$ is its total input. One key feature of capsule networks is that their many dimensions can keep specified information on an object such as location, orientation, etc. This information tends to be lost when max pooling in convolutional neural networks thus giving caps nets an edge in this regard.

### 2.3 Hierarchy of Parts

Now let’s consider how capsule networks handle a hierarchy of parts consider two new objects a boat and a house that are constructed with the rectangle and triangle objects in certain orientations like so:

![Figure 5: Secondary level house and boat objects (Géron 2017).](image-url)
The trick here is to utilize the output of the triangle and rectangle capsules (we call these lower level capsules primary capsules) to figure out if they are part of the secondary house or boat capsules. Essentially all the capsules in the primary layer try to predict the outputs of the capsules within the secondary layer. For example let’s consider the original boat capsule that was rotated by 16 degrees. Based on this orientation the rectangle capsule can either be a part of a boat rotated 16 degrees or a house rotated at 16 degrees. To make this prediction the rectangle capsule computes the dot product of a transformation matrix $W_{ij}$ with its own activation vector $u_i$:

$$\hat{u}_{ij} = W_{ij} u_i$$

Next, the triangle capsule does the same process and predicts the positions of a house and boat capsule. Based on the position of the triangle capsule it predicts that its either part of an upside down house or a boat rotated at 16 degrees.

Thus because the triangle and rectangle capsules agree on an orientation for a boat they conclude that they are indeed both parts of a boat object. This overall process is called routing by agreement, an algorithm that truly separates convolutional neural networks from capsule networks.
2.4 Routing By Agreement

Imagine that we have several different lower level capsules that all predict the presence of a secondary level capsule thus giving us a possible cloud of predictions vectors like so.

We then compute the mean of all these vectors and update the weights of each vector due to its distance relative to the mean (closer vectors get a higher weight).

We can now update the mean based on these higher weighted capsules and repeat this process multiple times until we reach a desired prediction weight.
This algorithm is similar to the K-means Clustering algorithm (Dabbura 2019). This is how we find clusters of agreement, now we can go over how the algorithm works in a bit more detail. We start by identifying the raw routing weights of the capsules $b_{ij}$ for all of the capsules and then apply a SoftMax function to this weight for each primary capsule. Thus giving our actual routing weight for each capsule.

Equation 3 (Sabour 2017).

$$b_{ij} = 0 \text{ for all } i, j$$

$$c_j = \text{softmax}(b_j)$$

We then compute a weighted sum $s_j$ for the predictions of the secondary capsule, this may give vectors larger than 1 so we squash them with the previously defined squash function in equation 1. This gives the first round actual outputs predictions for what secondary capsules the primary capsules might be a part of. We can then see which predictions were most accurate by estimating the scalar product of the predicted output vector $\hat{u}_{ji}$ and the actual product vector $v_j$ this is added to the capsules raw routing weight $b_{ij}$

Equation 4 (Sabour 2017).

$$b_{ij} += \hat{u}_{ji} \cdot v_j$$

The larger this number is the more likely what it predicted was right. We can repeat this process multiple times until we get a satisfactory value for $b_{ij}$ for a desired primary capsule. By doing this routing by agreement process we are able to retain more information on the objects and handle situations such as a crowded images with multiple houses and boats that make it hard for the computer to distinguish between them.
2.5 Applications and Other Capsule Network Details

As we have seen already Capsule Networks can help to create a nice image classifier in which objects within an image can be uniquely identified. This can help with more in depth applications such as facial recognition. Additionally as we will show, Capsule Networks can also help to solve several different Natural Language Processing tasks such as Part of Speech tagging. Other important details that the Hinton paper discussed includes applying a margin loss to the routing by agreement calculations in order to detect multiple classes within the image. Additionally Hinton also added a decoder network to their design in order to better reconstruct the input image without losing valuable information and overfit the image. Lastly one more detail about Capsule Network design is that it allows for the activation vectors to be quite interpretable as their many dimensions can represent various details about an object such as scale, thickness, localized part, etc. shown within the image below (Géron 2017).

<table>
<thead>
<tr>
<th>Scale and thickness</th>
<th><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Localized part</td>
<td><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></td>
</tr>
<tr>
<td>Stroke thickness</td>
<td><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></td>
</tr>
<tr>
<td>Localized skew</td>
<td><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></td>
</tr>
<tr>
<td>Width and translation</td>
<td><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></td>
</tr>
<tr>
<td>Localized part</td>
<td><img src="Sabour2017" alt="Interpretability of capsule dimensions" /></td>
</tr>
</tbody>
</table>
3 Part of Speech Tagging

Within Linguistics Part of Speech tagging also known as POS tagging “is the process of marking up a word in a corpus to a corresponding part of a speech tag.” (Aiswarya 2018).

Essentially we take individual words within a sentence and assign a part of speech to it, these parts of speech can be coarse tags such as a noun, verb, etc. or more fine grained such as gender, case, number, etc. For example consider the following sentence “The Dog ran” after running it through a POS tagger we might get the following tags for this sentence.

![Figure 11 Part of Speech tagger example.]

As we have previously discussed, Part of Speech tagging isn’t the solution to one particular problem but rather is a prerequisite to many other problems such as text to speech conversion or word sense disambiguation (Malhotra, Godayal 2018). Developing a Part of Speech tagger is no easy task one of the main challenges with this revolves around ambiguity.

In order to illustrate ambiguity let’s consider the sentence “flies like rice” this phrase can have two completely different meanings depending on surrounding words. For example, the word “flies” could be a noun if the sentence talks about insects or a verb if it talks about flying. A good working Part of Speech tagger would be able to discern these two labels based upon what it learns from the words around this phrase. Therefore due to these many nuances within language it’s hard to teach a machine perfect part of speech tagging.
So how do we go about measuring the correctness of a particular part of speech tagger? We use the metrics known as precision, recall and F1. Precision “is defined as the number of True Positives divided by the total number of positive predictions. It is also called the Positive Predictive Value” while Recall is “defined as the total number of True Positives divided by the total number of positive class values in the data. It is also called Sensitivity or the True Positive Rate” (Ramachandran 2018). When we test a POS tagger we might get a confusion matrix for its predicted outputs like so:

![Confusion Matrix](image)

**Figure 12** confusion matrix of POS taggers predicted outputs vs actual outputs (Shung 2020).

As we have described according to this table Precision = TP / TP + FN and Recall = TP / TP + FN. One final measurement that we need to familiarize ourselves with revolves around F1 score which is a function of Precision and Recall seeking an overall balance between the two and is defined as:

\[
F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}
\]

**Equation 5** (Shung 2020).

What’s the difference between F1 score and accuracy then? F1 scores essentially keep track of false positives and false negatives which give a better rounded distribution of how well a certain Part of Speech tagger functions.
In the end several implementations of Part of Speech tagging have been produced throughout the years, all of them do sequence modeling where words are tagged depending on the words within their context. For example many implementations utilize a Bi-LSTM, which is another architecture of a neural network (Wang 2015). Even well-known Natural Language Groups such as the Stanford Natural Language Processing Group have their own implementation through utilization of many java libraries (2018). However there has been little research with using Capsule Networks to perform Part of Speech tagging tasks.

4 Discussion

The main challenge with this research centered around how we could combine the capabilities of a Capsule Network which have shown promising results for image recognition and adjust the architecture to work as a Part of Speech tagger. There have been several other papers on capsule networks for NLP tasks but none that specifically combine Capsule Networks with Part of Speech tagging. For example Li et. al utilize Capsule Networks for a Chinese Word Segmentation task (2018). Others utilized Capsule Networks for sentiment analysis and genre classification (Stephan-It 2019; Zhao 2019). However, there was little research utilizing Capsule Networks for a Part of Speech tagging tasks and so figuring out where to begin was a big challenge.

4.1 Related Work Search

We began by looking into prior implementations of a capsule network and found many of them utilized TensorFlow and were designed based on the original Hinton paper for simple image recognition tasks (Sabour 2017). The real challenge came from finding an implementation that was built for a Natural Language Processing tasks. Many of these existing NLP
implementations offered solutions for genre classification but these didn’t align too well with our Part of Speech tagging goals. One of the main aspects we looked for was if an implementations primary and secondary capsules could be altered to look for fine-grained and coarse-grained Part of Speech tags. Yeung et al. offered a promising solution that implemented a capsule network for named entity recognition (2019). The way that this paper handled their data and implemented capsules aligned closely with our goals so that we could adjust it for POS tagging.

4.2 Data / Adapting Software

The original paper compared the outputs of a capsule network against that of a CNN model on the CNLL-2003 shared task and achieved better results with their capsule network architecture.


Figure 13: Example sentence from the CNLL-2003 shared task which assigns Named Entity Recognition tags to words in a sentence instead of Part of Speech tags (Language-Independent Named Entity Recognition (II) 2005).

This data’s input format was [word, POS tag, chunk tag, NER tag] with the chunk tag ignored. This made it easy for us to put our own data in as we could change the POS tag and NER tag sections with fine grained POS tag and coarse grained POS tag making the data format look like [word, fine grained POS tag, chunk tag, coarse grained POS tag]. We altered the Spanish Wikipedia Corpus and put it in this format for our input data files. With this format the fine grained POS tags such as gender, case, reflex, etc. were utilized as primary capsules to predict the output of the secondary capsules coarse grained tag for a word. We also added a new file called glove_helper.py in order to acquire the proper glove embeddings for the dataset.
4.3 Results

We trained the following data for 1 epoch on both a Capsule Network and a Convolutional Neural Network (CNN) to see how the each performed against one another our best models gave the following results:

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapsNet no features</td>
<td>96.91</td>
<td>96.67</td>
<td>96.78</td>
</tr>
<tr>
<td>CapsNet with features</td>
<td>99.32</td>
<td>99.22</td>
<td>99.27</td>
</tr>
<tr>
<td>CNN no features</td>
<td>95.12</td>
<td>95.58</td>
<td>95.38</td>
</tr>
<tr>
<td>CNN with features</td>
<td>98.45</td>
<td>98.42</td>
<td>98.44</td>
</tr>
</tbody>
</table>

Table 1: Results of Capsule Network vs CNN on Spanish Wikipedia Corpus.

Four models are shown, the ones without features only use the given word to predict its POS tags while the ones with features uses more information including the word and fine-grained tags to predict its POS tag. As shown these results reveal that the Capsule Network performed better than the CNN model possibly due to “over compression of local, token level information by max pooling into scalars” (Yeung 2019). In general it appears that routing by agreement preserved more useful information about a word then the CNN models max pooling. What’s also interesting centers around the performance of these models, overall they gave really promising scores with little training. This details to some of the pros of capsule networks against
CNNs that Hinton referenced within his paper such as Capsule Networks requiring less data, and preserving valuable information while training. In the end these results show an example of a Capsule Networks architecture outperforming the already established CNN architecture.

### 4.4 Low Resource Scenarios

As discussed one of the benefits of Capsule Networks is that they require less data to perform efficiently. This is a big benefit in terms of deep learning as less data allows for less time to train, additionally its overall easier to alter data and make it more interpretable if there’s less of it. We trained the no feature models but with 100%, 50% and 10% of the Spanish Wikipedia Corpus data while also changing the way the train/dev/test partitions were made giving us the following results:

<table>
<thead>
<tr>
<th>Model</th>
<th>Size of Data</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapsNet no features</td>
<td>100%</td>
<td>95.89</td>
<td>94.77</td>
<td>95.33</td>
</tr>
<tr>
<td>CNN no features</td>
<td>100%</td>
<td>93.45</td>
<td>93.85</td>
<td>93.65</td>
</tr>
<tr>
<td>CapsNet no features</td>
<td>50%</td>
<td>92.75</td>
<td>92.13</td>
<td>92.44</td>
</tr>
<tr>
<td>CNN no features</td>
<td>50%</td>
<td>89.26</td>
<td>88.79</td>
<td>89.02</td>
</tr>
<tr>
<td>CapsNet no features</td>
<td>10%</td>
<td>84.58</td>
<td>86.41</td>
<td>85.49</td>
</tr>
<tr>
<td>CNN no features</td>
<td>10%</td>
<td>79.54</td>
<td>78.05</td>
<td>78.79</td>
</tr>
</tbody>
</table>

*Table 2: Results of no feature Caps net and CNN models on various data sizes.*
As we can see the Capsule Network had a percentage loss in F1 score of about 3.03% between the 100% and 50% size of data and a 10.32% loss between 100% and 10%. This is better than the 4.94% and 15.87% loss in F1 score by the CNN for the same data sizes. What’s interesting about this is that the capsule network still performs quite well even with these cuts in data giving F1 scores of 92.44 and 85.49 at 50% and 10% data respectively. Overall showing that even with limited data the architecture is still capable of learning effectively and can still be useful in such low resource scenarios.

4.5 Future Work and Conclusion

Further research upon this topic can look at how we could make the capsules more interpretable so that the dimensions relate to certain aspects of a word such as word length, or surrounding words. This could be done by adding a classifier before routing by agreement begins so that the outputs of a primary capsule is first classified before deciding its relationship with a secondary capsule. Giulianelli et al. explore this by influencing the course of an LSTM when processing difficult sentences through the use of a classifier (2018). This would overall improve error analysis as we could monitor which specific words a capsule network failed to tag.

Within this thesis, we have explored a new type of neural network called a Capsule Network and shown its capabilities for a Part of Speech tagging task. This architecture was proposed in 2017 for image recognition and so a lot of previous work on Capsule Networks have explored its capabilities for that specific field (Sabour 2017). There is little research on Capsule networks applications within Natural Language Processing and to our knowledge no other papers have analyzed a Capsules Networks performance within a Part of Speech tagging task. As we have discussed previously Capsule Networks are a new type of architecture and so any research on the topic helps us further understand its possible applications. Therefore, what we have shown
is only the beginning in a wide spectrum of research that applies Capsule Networks within an
Natural Language Processing setting and possibly within other domains as well. We won’t know
for sure if this neural particular architecture will help in all circumstances but its implementation
and algorithms such as routing by agreement allow for much needed information to be retained
within a data set, overall giving it an edge when compared to other neural networks. In the end,
we hope that this exciting new development in deep learning will continue to be researched and
have beneficial impacts for society through its potential applications.
References


