

Please note that the following pages represent the translation of Chapter 3 of my MSc thesis, which was originally written in Persian. The full version of the thesis is available after the translation.

## Chapter 3

### 1. Methodology

#### 1.1. Hypotheses

In this chapter, considering the hypotheses that will be presented, the methodology of interest will be introduced. In this section, three fundamental hypotheses are proposed; the first hypothesis is that each sentence can be transformed into its constituent parts and structure. To create the constituent parts of the sentence, we represent it in a tree structure. To do this, we use the Stanford Parser tool. This tool was introduced by Klein and his colleagues in 2003 [7]. The tree structure of the sentence refers to parsing the sentence into its constituent words. This tree is binary, where each leaf represents a word of the sentence, and each parent node represents a phrase formed by the combination of two child words. In Figure 3-1, you can see the parsed tree structure of the sentence "There are slow and repetitive parts in the movie."

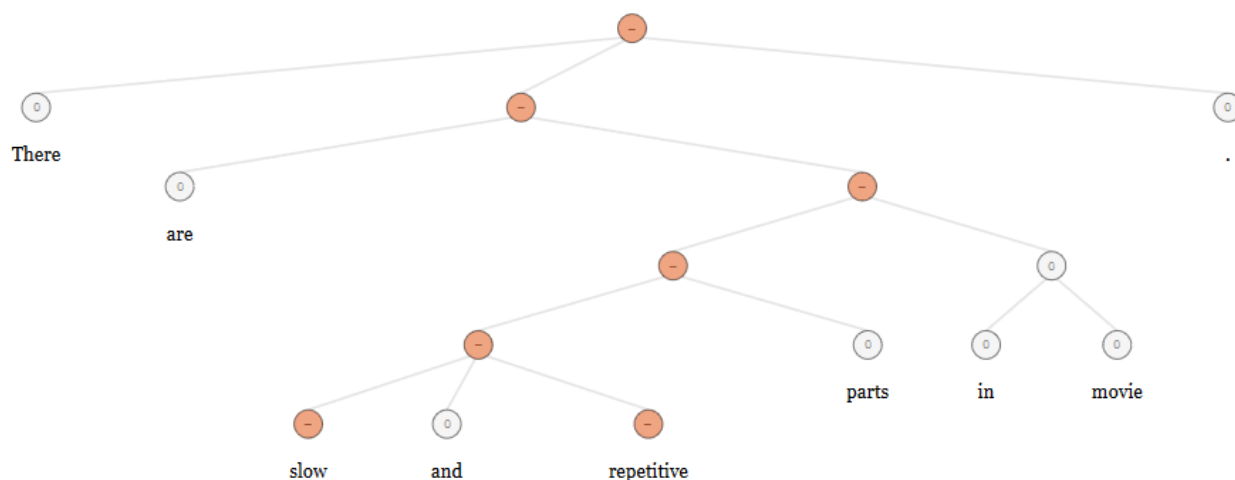


Figure 3-1: A sentence in a parsed (tree) form representation

The second hypothesis can be considered regarding the change in the efficiency of the model when there is a change in its parameters; in other words, by changing the network architecture and the parameters of word combinations, we will have a change in the performance of machine labeling. This hypothesis can be observed in the change of parameters from the RNN model to the MV-RNN, which has resulted in a positive change in the model's outcome. The third hypothesis, titled the performance of recursive networks in labeling words, phrases, and ultimately sentences, is known. In other words, the labeling of phrases resulting

from the combination of several words can be recognized by recursive neural networks. This hypothesis is acceptable considering that if the inputs to the network are not of variable size, the combination of two words should be such that the representation of the resulting phrase has the same size as the representation of a single word so that it can be reused and combined with another word and re-entered into the network.

In summary, in this section, three hypotheses have been presented as follows:

- Each sentence can be broken down into its components and structure.
- Changes in the network architecture and word combination parameters affect the performance of machine labeling.
- Phrases resulting from the combination of multiple words can be recognized by recursive neural networks.

## **2. Applied Methodology**

Neural networks have been utilized as powerful approximators in many fields [1]. The architecture of these networks is based on experience, and there are no predefined principles and rules for them [2]. In this section, we intend to introduce a type of neural network architecture. In this type of architecture, each input value of a neuron is combined with the adjacent neuron, determining the output of the next neuron. We refer to this type of architecture as Recursive Neural Networks (RNNs) [2]. Recursive neural networks are a type of neural network with a specific architecture that can be used for processing structured and sequential data, especially data represented by a directed acyclic graph. Deep network architectures consist of multiple nonlinear layers for data processing. Therefore, these types of networks are capable of combining meaningful data and extracting abstract meanings from them. recursive neural networks can be considered as an extension of recursive Neural Networks (RNNs), which have a specific type of slanted tree structure (as shown in Figure 3-2).

These types of networks have been used in applications such as parsing [4], sentence-level sentiment analysis [5] [6], etc. By having a structured representation of a sentence, such as its parsing tree, these types of networks calculate the representation of intermediate nodes of the tree based on the values of their child nodes using a bottom-up approach. By obtaining the values of the intermediate nodes and calculating the values of the higher nodes, the representation of a sentence can be obtained using this approach and used for tasks such as classification [5].

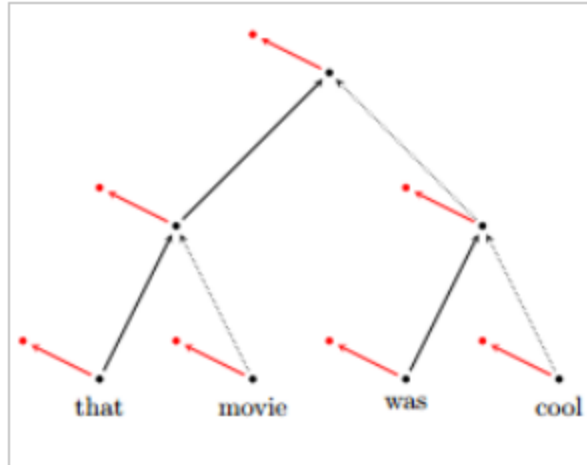


Figure 3-2: Structure of a Recursive Neural Network

Recursive neural network architectures have a series of shared weights that are recursively applied in a structured state: given a directed acyclic graph, this network sequentially traverses the nodes of this graph topologically and recursively applies changes to the nodes to obtain the representation of intermediate nodes from the representation obtained from the child nodes. As shown in Figure 3-3, in fact, Recursive Neural Networks are a simple recursive neural network with a special structure.

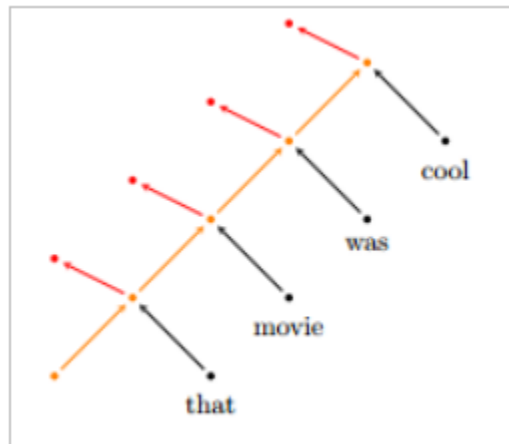


Figure 3-3- Structure of a Recursive Network

Even if recursive neural models can be used for any directed acyclic tree, here we only consider binary trees [5]. A given binary tree has an initial representation, for example, the parsing tree of a sentence with the representation of word features in its leaves, the recursive neural network calculates the representation of each internal node  $\eta$  as follows.

Formula 2-1

$$X_{\eta} = f(W_L x_{l_{\eta}} + W_R x_{r_{\eta}} + b)$$

How to calculate the internal nodes in recursive networks (the combination function of two child nodes and the production of the parent node) where  $l(\eta)$  and  $r(\eta)$  are the left and right children of node  $\eta$ ,  $W_L$  and  $W_R$  are the weight matrices connecting the left and right children to their parent, and  $b$  is the bias vector.  $W_L$  and  $W_R$  are equal matrices for all nodes, and the difference is whether  $l(\eta)$  and  $r(\eta)$  are leaf nodes or not. In this definition, one interpretation is of particular interest: the representation of the initial vectors of the children and parents in space is done with the same size. In the example of parsing a sentence tree, a recursive tree, by a combination function, combines the representation vectors of several words with each other to obtain the representation vector of a phrase containing several words in the same semantic space. After obtaining the representation vector of a node, in another step, the representation of the node is transformed into the output by the following formula.

Formula 2-2

$$y_{\eta} = g(U x_{\eta} + c)$$

Convert word vector to class label

In which,  $U$  is the output weight matrix in the space ( $c$  is the number of classes and  $d$  is the length of the node representation vector) and  $C$  is the Bias vector. In the implementation of a supervised project, the output (node class label) is  $\eta$ , and supervision is performed in this layer. For example, in the sentiment analysis problem, the value of  $\eta$  can be the predicted label value for a given phrase by the subtree with the root  $\eta$ . Then, during the supervised learning process, the error between the predicted value and the actual value is calculated and propagated from the root to the leaves [14]. The difference between various recursive neural network models lies in how the parent representation vector of children, which is in space, is calculated, in a bottom-up approach. Next, we introduce different recursive neural network models.

### 2.1.1. Recursive Neural Tensor Network (RNTN):

One of the problems with matrix-vector recursive neural networks is that the number of model parameters increases based on the size of the vocabulary. Having a function that can combine the features of children with a fixed number of parameters is highly desirable and preferred. Modifying RNN networks is a very good idea for implementing such a combinatorial function. In standard RNN networks, feature vectors entered into the network are directly related to each other by nonlinear functions. Generally, it seems that more interaction between input vectors will increase the accuracy of the model.

Based on the ideas presented, several questions arise: Can a powerful combination function be created that performs better than previous functions and combines smaller components of an expression more meaningfully with each other? To answer these questions, in this section, a

model called the Recursive Neural Tensor Network is introduced. The main idea is to use a tensor-based combination function for all nodes.

In Figure 3-4, you can see a tensor layer. The output of tensor multiplication,  $h \in \mathbb{R}$ , is defined by the following equation. But the finer labeling of each piece is as follows.

$$h = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix}; h_i = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[i]} \begin{bmatrix} b \\ c \end{bmatrix}.$$

Which  $V^{[1:d]} \in \mathbb{R}^{2d \times 2d \times d}$  is a tensor that determines multiple forms of two-way linear.

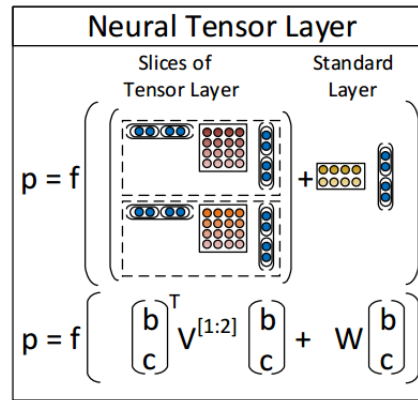


Figure3-4- A layer of Recursive Tensor Neural Network shows the hatched parts of a slice of the tensor layer

The RNTN combination function uses the following equation to calculate  $p_1$

which  $w$  was described in the previous model. The next parent vector, in a three-word expression, will be calculated with the same weights.

The main advantage of this model compared to the RNN model, which if we consider  $v$  equal to zero, the RNN model can be considered a special type of the RNTN model, is that the tensor can be directly related to the input vectors.

$$p_2 = f \left( \begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} \right).$$

Generally speaking, each piece of the tensor can be interpreted as representing a specific type of composition.

Adding another neural network layer can create a stronger combination function for RNTN. But initial experiences have shown that the optimization of this model is difficult and complex, and the vector interactions are more implicit than RNTN.

### **3. Summary**

In this chapter, algorithms based on recursive neural networks, which were used in text classification, were introduced. Formulas, relationships and parameters that should be changed during training were explained in detail and three models of recursion were introduced. These three models were RNN, MV-RNN and RNTN, each of which was completed or modified of the previous algorithm. Meanwhile, RNTN is the approach that we intend to compare with other introduced algorithms in this thesis. In the next chapter, practical results from the implementation of these algorithms will be presented.



دانشگاه علامه طباطبائی

دانشکده ریاضی و علوم رایانه

پایان نامه برای دریافت درجه کارشناسی ارشد

رشته علوم کامپیوتر – گرایش سیستم‌های هوشمند

تحلیل ترکیب متنی با استفاده از مدل درخت مفهومی جملات مبتنی بر شبکه عصبی ژرف

استاد راهنما

دکتر فرزاد اسکندری

استاد مشاور

دکتر محمدرضا اصغری اسکویی

استاد داور

دکتر حسن رشیدی

پژوهشگر

حمید محمودآبادی

زمستان 1397

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



کلیه‌ی حقوق مادی و معنوی اعم از چاپ و تکثیر، نسخه‌برداری، ترجمه، اقتباس و ... از این پایان‌نامه  
برای دانشگاه علامه طباطبائی محفوظ است. نقل مطالب با ذکر منبع منعی ندارد.

فرم اطاعات پایان نامه

کتابخانه مرکزی دانشگاه علامه طباطبائی

<p>عنوان به فارسی: تحلیل ترکیب متنی با استفاده از مدل درخت مفهومی جملات مبتنی بر شبکه عصبی ژرف</p>
<p>عنوان به انگلیسی:</p> <p>A Compositionality Text Analysis Using Tree structured sentences Based on Deep Neural Network</p>
<p>پژوهشگر : حمید محمودآبادی</p>
<p>استاد راهنما : دکتر فرزاد اسکندری</p> <p>استاد مشاور: دکتر محمدرضا اصغری اسکویی</p> <p>استاد داور: دکتر حسن رشیدی</p>
<p>مقطع تحصیلی: کارشناسی ارشد</p> <p>رشته تحصیلی: علوم کامپیوتر گرایش سیستم‌های هوشمند</p>
<p>کلیدواژه‌ها به زبان فارسی:</p> <p>شبکه عصبی، پردازش زبان طبیعی، تجزیه و تحلیل احساس، یادگیری ژرف، ساختار درختی جملات، قطبیت</p>
<p>کلیدواژه‌ها به زبان انگلیسی:</p> <p>Neural Network, Natural Language Processing, Sentiment Analysis, Deep Learning, Sentences Tree Structure, Polarity</p>

### چکیده به زبان فارسی:

در سال‌های اخیر استفاده از فضای معنایی کلمات، کاربرد بسیاری پیدا نموده است در برخی کاربردها که کلمات به صورت پشت‌سرهم و به صورت یک عبارت ظاهر می‌شوند، فضای معنایی کلمات نمی‌توانند معنای آنها را برسانند. در چند سال گذشته تلاش‌های بسیاری برای ترکیب معنای کلمات به‌طوری‌که معنای یک عبارت را به‌درستی برساند انجام شده است. در کاربردهایی مانند تحلیل نظرات کاربران، این مسئله نمود بیشتری پیدا کرده و نیاز به داشتن مدل بانظارت با قابلیت ترکیب معنای کلمات بالا را بیشتر از پیش می‌کند. جهت برداشتن گامی در جهت ساختن مدل عنوان شده، در این پایان‌نامه سعی شده تا مجموعه داده‌ای شامل 215154 عبارت و 11855 جمله معرفی شود. در این پایان‌نامه، مدلی با نام شبکه تنسوری عصبی بازگشتی معرفی شده است که سعی دارد چالش مطرح شده را با مجموعه داده‌ی معرفی شده حل نماید. این مدل بیشترین دقت را در بین مدل‌های معرفی شده دارا می‌باشد و باعث شده میزان دقت طبقه‌بندی جملات به دو طبقه‌ی مثبت و منفی را از 80٪ به 85.4 درصد افزایش دهد.

### چکیده به زبان انگلیسی:

Semantic word spaces have been very useful but cannot express the meaning of longer phrases in a principled way. Further progress towards understanding compositionality in tasks such as sentiment detection requires richer supervised training and evaluation resources and more powerful models of composition. To remedy this, we introduce a Sentiment Treebank. It includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences and presents new challenges for sentiment compositionality. To address them, we introduce the Recursive Neural Tensor Network. When trained on the new treebank, this model outperforms all previous methods on several metrics. It pushes the state of the art in single sentence positive/negative classification from 80% up to 85.4%.

صحت اطلاعات این فرم را براساس محتوای پایان نامه گواهی می‌کنم.

نام و نام خانوادگی: حمید محمودآبادی

امضاء







به نام خدا

## منشور اخلاق پژوهش

با یاری از خداوند سبحان و اعتقاد به اینکه عالم محضر خداوند است و همواره ناظر به اعمال انسان و به منظور پاس‌داشت مقام بلند دانش و پژوهش و نظر به اهمیت جایگاه دانشگاه در اعتلای فرهنگ و تمدن بشری ما دانشجویان دانشکده‌های دانشگاه علامه طباطبائی متعهد می‌گردیم اصول زیر را در انجام فعالیت‌های پژوهشی مد نظر قرار داده و از آن تخطی نکنیم:

- 1- اصل حقیقت‌جویی: تلاش در جهت حقیقت و وفاداری به آن و دوری از هرگونه پنهان‌سازی.
- 2- اصل رعایت حقوق: التزام به رعایت کامل حقوق پژوهشگران و پژوهیدگان و سایر صاحبان حق.
- 3- اصل مالکیت مادی و معنوی: تعهد به رعایت کامل حقوق مادی و معنوی دانشگاه و کلیه همکاران پژوهش.
- 4- اصل منافع ملی: تعهد به رعایت مصالح ملی و در نظر داشتن پیشبرد و توسعه کشور در کلیه مراحل پژوهش.
- 5- اصل رعایت انصاف و امانت: تعهد به اجتناب از هرگونه جانب‌داری غیر علمی و حفاظت از اموال، تجهیزات و منابع در اختیار.
- 6- اصل رازداری: تعهد به صیانت از اسرار و اطلاعات محرمانه افراد، سازمان‌ها و کشور و کلیه افراد و نهادهای مرتبط با تحقیق.
- 7- اصل احترام: تعهد به رعایت حریم‌ها در انجام تحقیقات و رعایت جانب نقد و خودداری از هرگونه حرمت‌شکنی.
- 8- اصل ترویج: تعهد به رواج دانش و اشاعه نتایج تحقیقات و انتقال آن به همکاران علمی و دانشجویان به غیر از مواردی که منع قانونی دارد.
- 9- اصل برائت: التزام به برائت‌جویی از هرگونه رفتار غیرحرفه‌ای و اعلام موضع نسبت به کسانی که حوزه علم و پژوهش را به شائبه‌های غیرعلمی می‌آلاینند.

نام و نام‌خانوادگی

تاریخ و امضا



### تعهد نامه‌ی اصالت پایان نامه

اینجانب حمید محمودآبادی دانش آموخته مقطع کارشناسی ارشد در رشته علوم کامپیوتر گرایش سیستم‌های هوشمند که در تاریخ 1397/11/13 از پایان نامه خود تحت عنوان "تحلیل ترکیب متنی با استفاده از مدل درخت مفهومی جملات مبتنی بر شبکه عصبی ژرف" با کسب نمره 17.75 دفاع نموده ام بدینوسیله متعهد می شوم:

- 1) این پایان نامه حاصل تحقیق و پژوهش انجام شده توسط اینجانب بوده و در مواردی که از دستاوردهای علمی و پژوهشی دیگران (اعم از پایان نامه، کتاب، مقاله و ...) استفاده نموده ام، مطابق ضوابط و رویه موجود، نام منبع مورد استفاده و سایر مشخصات آنرا در فهرست مربوطه ذکر و درج کرده ام.
- 2) این پایان نامه قبلاً برای هیچ مدرک تحصیلی (هم سطح، پایین تر یا بالاتر) در سایر دانشگاه ها و موسسات آموزش عالی ارائه نشده است.
- 3) چنانچه بعد از فراغت از تحصیل، قصد استفاده و هرگونه بهره برداری اعم از چاپ کتاب، ثبت اختراع و ... از پایان نامه داشته باشم، از حوزه معاونت پژوهشی واحد مجوزهای مربوطه را اخذ نمایم.
- 4) چنانچه در هر مقطع زمانی خلاف موارد فوق ثابت شود، عواقب ناشی از آن را می پذیرم و دانشگاه علامه طباطبایی مجاز است با اینجانب مطابق ضوابط و مقررات رفتار نموده و در صورت ابطال مدرک تحصیلی ام هیچگونه ادعایی نخواهم داشت.

نام و نام خانوادگی:

تاریخ و امضاء

تایید پایان نامه برای دریافت درجه‌ی کارشناسی ارشد توسط دانشجو

**عنوان پایان‌نامه:** تحلیل ترکیب متنی با استفاده از مدل درخت مفهومی جملات مبتنی بر شبکه

عصبی ژرف

نام دانشجو: حمید محمودآبادی

شماره دانشجویی: 9513139209

استاد راهنما: دکتر فرزاد اسکندری

اینجانب حمید محمودآبادی دانشجوی کارشناسی ارشد رشته‌ی علوم کامپیوتر گرایش سیستم‌های هوشمند دانشکده‌ی علوم ریاضی و رایانه دانشگاه علامه طباطبائی گواهی می‌کنم پژوهش‌های ارائه شده در پایان نامه با عنوان مذکور توسط شخص اینجانب انجام شده و درستی مطالب نگارش یافته، مورد تایید می‌باشد. همچنین تعهد می‌نمایم مطالب مندرج در این پایان‌نامه تاکنون برای دریافت هیچ نوع مدرک یا امتیازی توسط اینجانب یا فرد دیگری در هیچ کجا ارائه نشده است و در نگارش متن پایان‌نامه، شیوه‌ی نگارش مصوب دانشکده‌ی علوم ریاضی و رایانه را به‌طور کامل رعایت نموده‌ام. چنانچه در هر زمان، خلاف آنچه گواهی نموده‌ام مشاهده گردد، خود را از آثار حقیقی و حقوقی ناشی از دریافت مدرک کارشناسی ارشد محروم می‌دانم و هیچ‌گونه ادعایی نخواهم داشت.

امضا دانشجو:

زمستان 1397





دانشگاه علامه طباطبائی

دانشکده علوم ریاضی رایانه

پایان نامه برای دریافت درجه کارشناسی ارشد

تحلیل ترکیب متنی با استفاده از مدل درخت مفهومی جملات مبتنی بر شبکه

عصبی ژرف

پژوهشگر: حمید محمودآبادی

امضاء

استاد راهنما: دکتر فرزاد اسکندری

امضاء

استاد مشاور: دکتر محمدرضا اصغری اسکویی

امضاء

استاد داور: دکتر حسن رشیدی

امضاء

نماینده تحصیلات تکمیلی: دکتر حسن رشیدی

تقديم به بهترين داشته‌هايم، پدر و مادرم...

سپاس‌گزاری:

سپاس خدای را که هر توفیقی در گرو عنایات اوست. از زحمات استادان بزرگوارم جناب آقای دکتر فرزاد اسکندری و جناب آقای محمدرضا اصغری اسکویی که با راهنمایی‌های بی‌دریغشان در به نتیجه رسیدن این پایان‌نامه مرا یاری نمودند، کمال تشکر را دارم. همچنین از استاد بزرگوار، جناب آقای دکتر حسن رشیدی که زحمت داوری این رساله را متقبل شدند، تشکر و قدردانی را می‌نمایم.

در پایان از خانواده‌ام، به ویژه پدر و مادرم که با حمایت‌های خویش، همواره مرا پشتیبانی کرده‌اند، نهایت سپاس و قدرشناسی را دارم.

حمید محمودآبادی

پاییز 1397

<b>1 کلیات پژوهش</b>	<b>19</b>
1.1 مقدمه	19
1.2 تعریف مسئله	20
1.3 اهمیت مطالعه	21
1.4 کاربردهای مطالعه انجام شده	22
1.5 هدف از مطالعه و پرسش‌های پژوهش	23
1.6 ساختار پایان‌نامه	24
<b>2 مروری بر ادبیات و کارهای انجام شده</b>	<b>26</b>
2.1 تعریف واژگان کلیدی	26
2.2 مروری بر کارهای انجام شده	28
2.2.1 شبکه‌های بازگشتی	33
2.2.1.1 شبکه‌های عصبی بازگشتی:	37
2.2.1.2 شبکه‌های عصبی بازگشتی ماتریس-بردار:	38
2.3 چالش‌ها	40
2.4 ابزارهای قابل به کارگیری	40
2.5 نتیجه‌گیری	41
<b>3 متدولوژی</b>	<b>43</b>
3.1 فرضیه‌ها	43
3.2 متدولوژی به کار برده شده	45
3.2.1 شبکه تنسور عصبی بازگشتی (RNTN):	49
3.3 جمع بندی	51
<b>4 پیاده‌سازی و ارزیابی</b>	<b>53</b>
4.1 مجموعه داده‌ها	53
4.2 ارزیابی نمودارهای حاصل شده	55
4.3 توصیه‌ها و کارهای قابل انجام در آینده	67
4.3.1 پیاده‌سازی در زبان فارسی	67
4.3.2 ایجاد تغییر در معماری شبکه	68
4.3.3 استفاده از معماری موجود جهت پیش بینی سری‌های زمانی	68
4.4 نتیجه‌گیری	69



شکل 1-1- روند انجام یک پروژه تحلیل نظر.....	20
شکل 2-1- ساختار کلی یک شبکه عصبی مصنوعی.....	27
شکل 1-2- روش‌های مختلف تجزیه و تحلیل متن [2].....	30
شکل 2-2- ساختار یک شبکه عصبی بازگشتی.....	34
شکل 3-2- ساختار یک شبکه Recurrent.....	35
شکل 4-2- درخت باینری یک عبارت سه کلمه ای.....	38
شکل 5-2- درخت تجزیه‌ی یک عبارت سه کلمه‌ای شامل ماتریس و بردار ویژگی.....	39
شکل 1-3- چگونگی نمایش یک جمله بصورت تجزیه شده (درختی).....	44
شکل 2-3- ساختار یک شبکه عصبی بازگشتی.....	46
شکل 3-3- ساختار یک شبکه Recurrent.....	47
شکل 4-3- یک لایه از شبکه تنسور عصبی بازگشتی، قسمت‌های خط‌چین یک تکه از لایه‌ی تنسوری را نمایش می‌دهد.....	50
شکل 1-4- رابط برچسب‌گذاری. عبارات بصورت تصادفی به برچسب‌گذاران نمایش داده می‌شد و آنها درجه‌ی مثبت یا منفی بودن آنها را مشخص می‌کنند [5].....	54
شکل 2-4- نمودار هیستوگرام نرمال شده‌ی برچسب‌های عبارات.....	55
شکل 3-4- نمودار دقت ریز طبقه‌بندی عبارات [6].....	57
شکل 4-4- نمودار به‌دست آمده از پیاده‌سازی مدل RNTN جهت طبقه‌بندی جملات.....	58
شکل 5-4- نمودار به‌دست آمده از پیاده‌سازی مدل RNTN جهت طبقه‌بندی کلمات.....	58
شکل 6-4- نمودارهای ماتریس سرگشتگی مدل RNTN جهت طبقه‌بندی جملات.....	59
شکل 7-4- یک مثال از پیش‌بینی صحیح مدل در جمله با ساختار ".... اما ... ".....	62
شکل 8-4- پیش‌بینی مدل قطبیت جملات مثبت و منفی و قطبیت معکوس شده‌ی آنها.....	63
شکل 9-4- تغییر فعالیت مدل‌ها با جملاتی که قطبیت آنها معکوس شده.....	65
شکل 10-4- میانگین تشخیص مثبت‌ترین و منفی‌ترین عبارات با طول‌های مختلف [10].....	66

جدول 4-1 - دقت پیش‌بینی مدل‌ها در حالت ریز طبقه‌بندی (5 طبقه) در دو سطح گره و جمله	60
جدول 4-2 - دقت پیش‌بینی مدل‌ها در حالت طبقه‌بندی باینری (2 طبقه) در دو سطح گره و جمله	61
جدول 4-3 - دقت مدل در تشخیص جملاتی که قطبیت آنها توسط یک کلمه معکوس شده است	64
جدول 4-4 - مثالی از مثبت‌ترین و منفی‌ترین عباراتی که مدل شناسایی کرده است [6]	66



# 1 کلیات پژوهش

## 1.1 مقدمه

امروزه متوجه شدن نظر مشتری راجع به یک کالا یا کالاهایی که در یک فروشگاه به فروش می‌رسد می‌تواند باعث بهبود کیفیت کالا و خدمات دهی شود. به همین دلیل، نظرات و نوشته‌هایی که در زیر محصولات یک سایت نوشته می‌شوند را می‌توان جزئی از نظرات کلی که کاربران دارند به حساب آورد. بنابراین تجزیه و تحلیل این نظرات و نوشته‌ها می‌تواند به‌طور مستقیم تاثیر مثبتی بر روی میزان فروش و بهبود کیفیت کالا یا خدمات داشته باشد. داشتن یک سیستم تحلیل نظرات کاربران می‌تواند به‌صورت کارآ، در انواع کسب‌وکارها به خصوص در کسب‌وکارهای اینترنتی مورد استفاده قرار بگیرد. دلیلی که باعث می‌شود یک سیستم برای این کار طراحی شود، تولید محتوای بسیار توسط کاربران کسب‌وکارهای اینترنتی می‌باشد، به‌طوری‌که خواندن، تحلیل و نتیجه‌گیری از آن توسط نیروی انسانی غیرقابل‌انجام می‌باشد. در گذشته روش‌ها و الگوریتمهای<sup>1</sup> بسیاری جهت انجام این مهم معرفی شده است. عدم کارآیی به‌علت دقت پایین الگوریتم‌های پیشنهادی باعث شده تا مطالعات در این حوزه بیشتر شود و هر روزه الگوریتم‌های جدیدتر با دقت بیشتری معرفی شوند. در این پایان‌نامه سعی شده است روش جدید معرفی گردد که دقت آن بیشتر از روش‌های قبلی می‌باشد. در ادامه این فصل، مسئله‌ای که سعی در حل آن داریم بیشتر تشریح می‌شود، سپس واژگان کلیدی تعریف می‌شوند و پس از آن اهمیت مطالعه و همچنین کاربردهای مطالعه‌ی انجام شده مورد بررسی قرار می‌گیرد. در پایان نیز چالش‌هایی که در انجام این پایان‌نامه با آن مواجه هستیم معرفی می‌گردند.

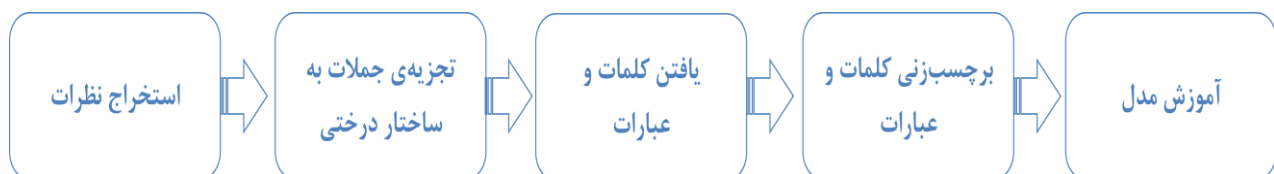
---

<sup>1</sup> Algorithms

## 1.2 تعریف مسئله

در تجزیه و تحلیل نظرات کاربران<sup>1</sup>، بزرگترین مسئله، متوجه شدن این است که کاربر نسبت به کالا یا محصولی که نظر داده، دیدگاه مثبتی دارد یا منفی. نظراتی که کاربران در زیر محصولات درج نموده‌اند، معمولاً شامل چند جمله کوتاه است مثلاً "فیلم خوبی نبود" یا "محصول کیفیت قابل قبولی دارد". تجزیه‌ی نظرات به جملات تشکیل دهنده‌ی آنها، و تقسیم جملات به نظرات مثبت و منفی می‌تواند به فهمیدن میزان رضایت کلی مشتری از محصول را برای ما آشکار کند. مسئله‌ی مهمی که در این قسمت با آن مواجه هستیم این است که چگونه مثبت یا منفی بودن یک جمله را متوجه شویم. مثلاً در مثال "فیلم خوبی نبود" که جمله‌ای منفی است با جمله‌ی "فیلم خوبی بود" که جمله‌ای مثبت است تنهای یک حرف الفبا اختلاف وجود دارد. متوجه شدن اینکه هنگام ترکیب عبارات و کلمات و ساخت جملات، روابط بین کلمه و عبارت چگونه بر مثبت یا منفی بودن جمله تاثیر می‌گذارد، خود مسئله‌ای است که در این پایان‌نامه به بررسی آن و ارائه راه حل برای آن پرداخته شده است. به عبارت دیگر ما در این پایان‌نامه به دنبال تابعی هستیم که تک تک کلمات یک جمله را گرفته، آنها را با هم ترکیب معنایی کند و در نهایت با توجه به ترکیبی که از کل کلمات جمله به دست آورده، مشخص کند که جمله مثبت بوده است یا منفی.

به‌طور کلی روندی که جهت تعیین این مهم انجام می‌شود در شکل 1-1 آورده شده است:



شکل 1-1- روند انجام یک پروژه تحلیل نظر

<sup>1</sup> Users sentiment analysis

در ادامه توضیح مختصری در مورد پنج گام فوق آورده شده است.

- استخراج نظرات: در این قسمت نظرات کاربران مربوط به یک سیاست، ویدئو یا یک محصول که در یک یا چند سایت وجود دارد، استخراج می‌شود.
- تجزیه‌ی جملات به ساختار درختی: در این مرحله، جملات به دست آمده از مرحله‌ی قبل با استفاده از ابزار تجزیه‌کننده<sup>۱</sup> به صورت ساختار درختی نمایش داده می‌شود.
- یافتن کلمات و عبارات: در این گام، برگ‌ها و گره‌های میانی درخت جملات جهت برچسب‌زنی جداسازی و آماده می‌شوند.
- برچسب‌زنی کلمات و عبارات: در این بخش، کلمات و عبارات برچسب‌دهی می‌شوند که هر برچسب نشان دهنده‌ی طبقه<sup>۲</sup> آن کلمه یا عبارت است.
- آموزش مدل: در این قسمت داده بدست آمده از مراحل قبل، وارد شبکه شده و شبکه طبق داده مورد نظر آموزش می‌بیند.

### 1.3 اهمیت مطالعه

همان‌طوری که در مقدمه‌ی این فصل اشاره شد، متوجه شدن نظر مشتری راجع به یک کالا یا کالاهایی که در یک فروشگاه به فروش می‌رسد می‌تواند باعث بهبود کیفیت کالا و خدمات‌دهی شود. به همین دلیل، نظرات و نوشته‌هایی که در زیر محصولات یک سایت نوشته می‌شوند را می‌توان جزئی از نظرات کلی که کاربران دارند به حساب آورد. بنابراین تجزیه و تحلیل این نظرات و نوشته‌ها می‌تواند به‌طور مستقیم تاثیر مثبتی بر روی میزان فروش و بهبود کیفیت کالا یا خدمات داشته باشد. داشتن یک سیستم تحلیل نظرات کاربران می‌تواند

---

<sup>1</sup> Parser

<sup>2</sup> Class

به صورت کارآ، در انواع کسب و کارها به خصوص در کسب و کارهای اینترنتی مورد استفاده قرار بگیرد. دلیلی که باعث می شود یک سیستم برای این کار طراحی شود، تولید محتوای بسیار توسط کاربران کسب و کارهای اینترنتی می باشد، به طوری که خواندن، تحلیل و نتیجه گیری از آن توسط نیروی انسانی غیر قابل انجام می باشد. از طرفی دیگر، نظراتی که کاربران در زیر محصولات درج نموده اند، معمولاً شامل چند جمله کوتاه است مثلاً "فیلم خوبی نبود" یا "محصول کیفیت قابل قبولی دارد". تجزیه ی نظرات به جملات تشکیل دهنده ی آنها، و تقسیم جملات به نظرات مثبت و منفی می تواند به فهمیدن میزان رضایت کلی مشتری از محصول را برای ما آشکار کند. مسئله ی مهمی که در این قسمت با آن مواجه هستیم این است که چگونه مثبت یا منفی بودن یک جمله را متوجه شویم. مثلاً در مثال "فیلم خوبی نبود" که جمله ای منفی است با جمله ی "فیلم خوبی بود" که جمله ای مثبت است تنهای یک حرف الفبا اختلاف وجود دارد. اهمیت این مطالعه هنگامی پدیدار میشود که بتوانیم، جملات را به درستی به دسته های مثبت و منفی تقسیم کنیم. در این صورت می توانیم درک صحیح از نظرات مشتریان داشته و تاثیر آن را در افزایش بازدید کنندگان سایت یا استفاده کنندگان از محصول بیابیم.

#### 1.4 کاربردهای مطالعه ی انجام شده

بدیهی ترین کاربردی که می توان برای این مطالعه متصور شد، تجزیه و تحلیل نظرات کاربرانی است که از یک کالا استفاده کرده اند و در مورد آن در یک سایت اینترنتی نظر داده اند. اگر دسته بندی و نمایش درستی از کلیات نظرات کاربران در یک سایت وجود داشته باشد، هنگام خرید یک محصول به سادگی می توانیم از نظراتی کاربرانی که قبلاً از محصول استفاده کرده اند را با یک نگاه به آن متوجه شویم و نیازی به اتلاف وقت و خواندن بخش زیادی از نظرات کاربران برای آگاه شدن از آنها نخواهیم داشت. از کاربردهایی دیگری که می توان برای تجزیه و تحلیل نظرات کاربران متصور شد، آگاه شدن از نظرات مردم یک کشور در قبال اتخاذ یک

سیاست خاص در آن کشور است. مردم به‌هنگام اتخاذ یک سیاست یا سرگیری یک رابطه با یک کشور دیگر، نظراتشان را در قالب پست در شبکه‌های اجتماعی مختلف، منتشر می‌کنند. با جمع‌آوری این نظرات و تحلیل آنها می‌توان بهترین سیاست را در برابر نظرات آنها اتخاذ نمود. در کل در هر قسمت از فضای مجازی که جملاتی از کاربران در مورد هر موضوعی نوشته شده باشد را می‌توان با این روش تحلیل نمود و براساس آن، تصمیم‌گیری کرد.

## 1.5 هدف از مطالعه و پرسش‌های پژوهش

یکی از اهداف انجام این مطالعه، شناخت و بررسی الگوریتم‌هایی است که برای تحلیل نظرات کاربران به کار می‌رود. هدف اصلی از انجام این مطالعه، ارتقا و افزایش دقت ابزارهای موجود می‌باشد که با کاهش خطا، باعث تصمیم‌گیری‌های درست در حوزه‌های تحلیل شده‌ی مرتبط شود. لذا برآنیم تا با مطالعه‌ی وضعیت مدل‌های موجود در حوزه‌ی تحلیل نظرات کاربران، به معرفی یکی از ابزارهای بهبود یافته بپردازیم. به‌طور کلی در این مطالعه در صدد آن هستیم که برای پرسش‌های عنوان شده در زیر پاسخی بیابیم.

1- آیا می‌توان با استفاده از شبکه‌های عصبی ژرف و ساختار درختی جملات، میزان قطبیت جملات را

تشخیص داد؟

2- کارایی تجزیه و تحلیل احساسات مبتنی بر شبکه عصبی ژرف در جملات انگلیسی چگونه است؟

3- معماری شبکه و پارامترهای تاثیرگذار در کارامدی برچسب زنی ماشینی چه تاثیری دارد؟

4- آیا می‌توان جملات را به ساختار درختی تشکیل دهنده آن تبدیلی نمود؟

## 1.6 ساختار پایان نامه

در ادامه‌ی این پایان نامه، ابتدا در فصل 2 به بررسی و مطالعه‌ی تاریخچه و پیشینه‌ی پژوهشی این مطالعه می‌پردازیم؛ در این فصل در ابتدا، پژوهش‌هایی که پیش‌تر از این مطالعه انجام شده است را مورد بررسی قرار می‌دهیم. سپس ابزارهایی که می‌توان جهت پیاده‌سازی این پژوهش مورد استفاده قرار داد را به صورت خلاصه مورد مقایسه قرار می‌دهیم و یک ابزار را بر اساس مزایا و نقاط قوت آن، جهت مدل‌سازی انتخاب می‌کنیم و در بخش نهایی، نتیجه‌گیری از مباحث مطرح شده در این فصل خواهیم داشت. در فصل 3 متدولوژی مورد استفاده جهت مدل‌سازی مورد بحث و بررسی قرار خواهد گرفت. و در نهایت در فصل 4 مجموعه داده‌ی مورد استفاده در مدل‌سازی معرفی می‌گردد. سپس نتایج مدل به دست آمده با استفاده از ابزار و مجموعه داده معرفی شده، مورد بررسی قرار می‌گیرد و در بخش آخر این فصل نتیجه‌گیری و توصیه برای کارهایی که در آینده قابل انجام است، ارائه خواهد شد.



## 2 مروری بر ادبیات و کارهای انجام شده

### 2.1 تعریف واژگان کلیدی

شبکه‌های عصبی مصنوعی<sup>۱</sup>: یک شبکه عصبی مصنوعی<sup>۲</sup>، از سه لایه ورودی، خروجی و پردازش تشکیل می‌شود. هر لایه شامل گروهی از سلول‌های عصبی (نورون<sup>۳</sup>) است که عموماً با کلیه نورون‌های لایه‌های بعدی در ارتباط هستند، مگر این که طراح ارتباط بین نورون‌ها را محدود کند؛ ولی نورون‌های هر لایه با سایر نورون‌های همان لایه، ارتباطی ندارند. نورون کوچک‌ترین واحد پردازشگر اطلاعات است که اساس عملکرد شبکه‌های عصبی را تشکیل می‌دهد. یک شبکه عصبی مجموعه‌ای از نورون‌هاست که با قرار گرفتن در لایه‌های مختلف، ساختار خاصی را بر مبنای ارتباطات بین نورون‌ها در لایه‌های مختلف تشکیل می‌دهند. نورون می‌تواند یک تابع ریاضی خطی یا غیرخطی باشد، در نتیجه یک شبکه عصبی که از اجتماع این نورون‌ها تشکیل می‌شود، نیز می‌تواند یک سامانه کاملاً پیچیده و غیرخطی باشد. در شبکه عصبی هر نورون به طور مستقل عمل می‌کند و رفتار کلی شبکه، برآیند رفتار نورون‌های متعدد است. به عبارت دیگر، نورون‌ها در یک روند همکاری، یکدیگر را تصحیح می‌کنند. شبکه‌های عصبی به دلیل دارا بودن ویژگی یادگیری آنلاین باعث شده است که در سیستم‌هایی که مدام در حال تغییر می‌باشند مورد استفاده قرار گیرند. شکل 1-2 ساختار کلی یک شبکه عصبی مصنوعی را نمایش می‌دهد.

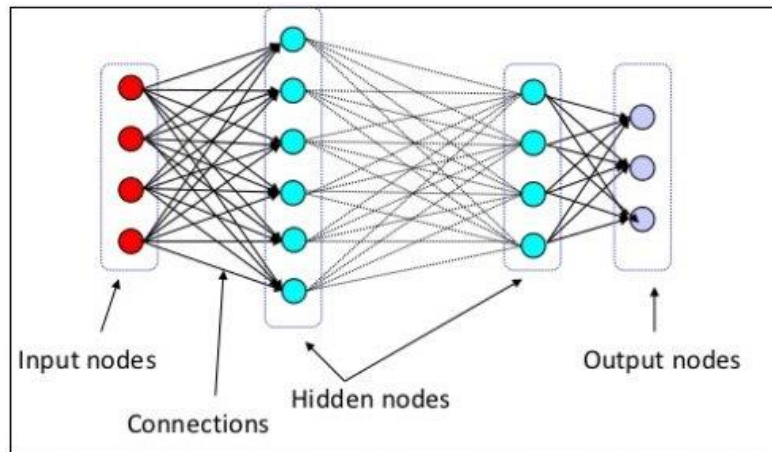
---

<sup>1</sup> Artificial Neural Networks (ANN)

<sup>2</sup> Artificial neural network

<sup>3</sup> Neuron





شکل 1-2 ساختار کلی یک شبکه عصبی مصنوعی

- **پردازش زبان طبیعی<sup>۱</sup>:** دانش زبان‌شناسی محاسباتی است که به تعامل بین کامپیوتر و زبان‌های (طبیعی) انسانی می‌پردازد؛ پس چالش اصلی و عمده در این زمینه درک زبان طبیعی و ماشینی کردن فرایند درک و برداشت مفاهیم بیان‌شده با یک زبان طبیعی انسانی است. به تعریف دقیق‌تر، پردازش زبان‌های طبیعی عبارت است از استفاده از رایانه برای پردازش زبان گفتاری و زبان نوشتاری.
- **تجزیه و تحلیل احساس<sup>۲</sup>:** منظور از تجزیه و تحلیل احساس، متوجه شدن بار معنایی یک متن مشتمل بر تعدادی جمله می‌باشد. حال این بار معنایی می‌تواند بر بار معنایی کلی جمله یا بار معنایی در مورد جنبه‌ی خاصی باشد.
- **یادگیری ژرف<sup>۱</sup>:** یک زیر شاخه از یادگیری ماشینی است که اساس آن بر یادگیری نمایش دانش و ویژگی‌ها در لایه‌های مدل است.

<sup>1</sup> Natural Language Processing (NLP)

<sup>2</sup> Sentiment Analysis (SA)

- ساختار درختی جملات<sup>۲</sup>: منظور از ساختار درختی جملات، چگونگی اتصال کلمات به یکدیگر و تشکیل جمله می‌باشد.
- قطبیت کلمات<sup>۳</sup>: قطبیت نشان دهنده‌ی میزان مثبت یا منفی بودن بار معنایی کلمه، عبارت یا جمله را نشان می‌دهد.

## 2.2 مروری بر کارهای انجام شده

تعیین قطبیت جمله را می‌توان به صورت طبقه‌بندی به دو دسته مثبت و منفی (طبقه‌بندی دودیی [3][4]) و یا به صورت طبقه‌بندی به چند کلاس انجام داد([5][6][7]). برای انجام این کار، از رویکردهای مختلفی بهره گرفته می‌شود:

- تحلیل در سطح متن: در این نوع تحلیل، کل متن را به صورت یک موجودیت در نظر گرفته می‌شود و رویکردهای تحلیل بر روی کل متن پیاده سازی می‌شود.

---

<sup>1</sup> Deep Learning (DL)

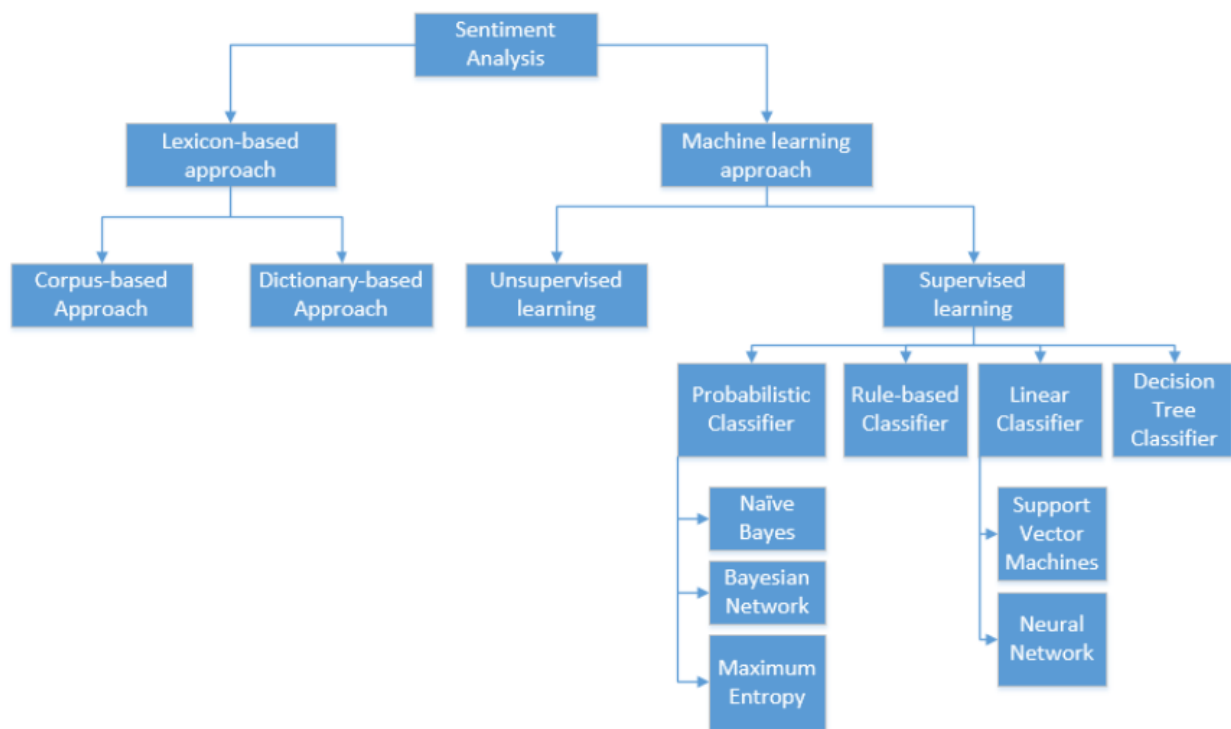
<sup>2</sup> Sentence Tree Structure

<sup>3</sup> Words Polarity

- **تحلیل در سطح جمله:** در این نوع تحلیل، هر جمله به صورت یک موجودیت در نظر گرفته شده و رویکرد تحلیل بر روی هر جمله پیاده میشود. سپس با خلاصه کردن تحلیل های حاصل شده از جملات، تحلیلی در مورد کل متن بدست می آید.
  - **تحلیل جنبه گرا:** در این نوع تحلیل، هر جمله را بر اساس یک ویژگی مد نظر، به طبقه های مختلف کیفی یا کمی تقسیم میکنیم. به عنوان مثال، برای یک گوشی تلفن همراه می توان ویژگی هایی مانند وزن، کیفیت صفحه، اندازه ی صفحه و ... را متصور شد و آنرا به طبقه های مختلف کیفی و کمی تقسیم بندی نمود.
- رویکردهای تجزیه و تحلیل متن را به دو دسته ی مبتنی بر یادگیری ماشین و مبتنی بر واژگان<sup>1</sup> تقسیم بندی می کنند [2]. در شکل 2-2 می توانید این تقسیم بندی را بهتر و با جزئیات بیشتر مشاهده کنید.

---

<sup>1</sup> Lexicon



شکل 2-2 روش‌های مختلف تجزیه و تحلیل متن [2]

در این قسمت قصد داریم تجزیه و تحلیل متن با استفاده از الگوریتم‌های یادگیری ماشین با نظارت<sup>۱</sup> را تشریح کنیم و از تشریح روش یادگیری بدون نظارت و روش‌های مبتنی بر واژگان به دلیل مربوط نبودن به روش انتخابی این پایان‌نامه صرف نظر می‌کنیم.

### روش‌های یادگیری ماشین با نظارت:

منظور از یادگیری ماشین با نظارت، روش‌هایی از الگوریتم‌های یادگیری ماشین است که جهت مدل‌سازی نیاز به داده‌های برچسب‌دار دارند. یکی از روش‌های یادگیری با نظارت، روش مبتنی بر کیسه کلمات<sup>۲</sup> می‌باشد. معمولاً این روش‌ها، یک سند را به عنوان مجموعه‌ای از ویژگی کلمات  $d = (W_1, W_2, W_3, \dots, W_N)$  می‌بینند،

<sup>۱</sup> Supervised

<sup>۲</sup> Bag of Words

به طوری که  $N$  مجموعه‌ای از کلمات یکتا<sup>1</sup> در مجموعه داده و  $W_i$  وزن آیتام می‌باشد، سپس آن سند را طبق همان کلمات تحلیل می‌کنند و در فهم رابطه بین کلمات ناتوان هستند [24]. جهت تبدیل مجموعه داده به بردار ویژگی‌ها، می‌بایست، دایره‌ی واژگان<sup>2</sup> از مجموعه داده‌ی آموزش استخراج می‌شد. علاوه بر این، هر مدل دیگری که می‌توانست بردار ویژگی ارائه دهد، قابل استفاده در این مدل‌ها بود و محدودیتی از این لحاظ وجود نداشت. مثال‌های مدل‌هایی که بردار ویژگی ارائه می‌دهند در زیر آورده شده است [24]:

- مدل ویژگی دودویی: اگر کلمه در سند وجود داشت، مقدار معادل آن در بردار برابر 1 و در غیر این صورت، برابر با 0 در نظر گرفته می‌شود.

- فراوانی لغت<sup>3</sup>: تعداد دفعاتی که یک لغت در سند ظاهر می‌شود را به عنوان ویژگی در نظر می‌گیرد.

- $TF-IDF$  (Term Frequency – inverse document frequency): این روش میزان اهمیت یک کلمه را در نظر می‌گیرد (روش فراوانی لغت فقط تعداد دفعات ظاهر شدن را در نظر می‌گرفت)

پس از اینکه سند به عنوان یک بردار ویژگی تعیین مشخص شد، می‌تواند توسط طبقه‌بندها<sup>4</sup> جهت مدل‌سازی مورد استفاده قرار بگیرد. انواع متدهای مختلف برای آموزش مدل می‌تواند مورد استفاده قرار بگیرد. در زیر روش‌های مختلف آموزش مدل معرفی شده است:

معمول‌ترین و ساده‌ترین روشی که برای طبقه‌بندی متن مورد استفاده قرار می‌گیرد، Naïve Bayes است [24][25][26][27]. این مدل مبتنی بر نظریه‌ی بیزین با فرض این که ویژگی‌ها مستقل از هم می‌باشند. این روش، میزان احتمال تعلق نمونه را به یک کلاس مشخص می‌کند. از مزایای این روش می‌توان به سادگی

---

<sup>1</sup> Unique

<sup>2</sup> Vocabulary

<sup>3</sup> Term frequency

<sup>4</sup> Classifiers

در پیاده‌سازی، سرعت بالای آموزش، و نتایج تقریباً مطلوب آن را نام برد [4][26][27]. مشکل این روش این بود که نمی‌توانست ویژگی‌هایی که در واقعیت وجود دارند را لحاظ کند.

روش دیگر طبقه‌بندی، روش مبتنی بر قانون<sup>1</sup> نام دارد. ایده اصلی این روش، اعمال قوانینی که توسط یک خبره تولید شده است بر روی متن بود. در مواردی که قوانین بسیار زیادی استفاده می‌شد، این روش نتایج مطلوبی داشت، اما ایجاد چنین قوانینی نیاز به صرف زمان بسیار و وجود یک فرد خبره داشت. این روش توسط Chikersal و همکارانش در [28] مورد استفاده قرار گرفت.

طبقه‌بند بردار پشتیبان ماشین<sup>2</sup> روش دیگری بود که در [25][26][27] مورد استفاده قرار گرفت. این مدل، در مواقعی که طبقه‌بندی باینری مد نظر است، در فضای ویژگی‌ها، به دنبال ابرصفحه‌ای است که تفکیک بین دو طبقه را با بیشترین حاشیه به وجود بیاورد. حاشیه، فاصله‌ی بین ابرصفحه تا نزدیک‌ترین نمونه کلاس در نظر گرفته می‌شود. نمونه‌هایی که به ابرصفحه نزدیک می‌باشند و حاشیه طبق فاصله‌ی آن دو معین می‌شود را بردار پشتیبان می‌نامند که برای طبقه‌بندی بسیار مهم می‌باشند چرا که انتخاب آنها در جای‌گیری ابرصفحه و در نهایت دقت طبقه‌بند تاثیر می‌گذارد. این روش برای مجموعه‌داده‌های بزرگ مناسب نبود چرا که پیچیدگی محاسباتی و زمانی بسیاری را در طی روند مدل‌سازی به سیستم تحمیل می‌کرد.

روش دیگری که برای طبقه‌بندی متن مورد استفاده قرار می‌گیرد، شبکه‌های عصبی مصنوعی می‌باشد. شبکه‌های عصبی، با تغییر وزن‌های خود قادر به یادگیری مدل‌های بسیار پیچیده می‌باشند. این روش، روش اصلی مورد استفاده در این پژوهش می‌باشد، در ادامه همین فصل توضیحات با جزئیات بیشتر در مورد این روش مطرح خواهد شد.

---

<sup>1</sup> Rule-Based approach

<sup>2</sup> Support Vectors Machine

## 2.2.1 شبکه‌های بازگشتی<sup>۱</sup>

شبکه‌های عصبی، به عنوان یک تخمین‌زننده<sup>۲</sup> قوی در بسیاری از زمینه‌ها به کار برده شده است [1]. معماری این شبکه‌ها براساس تجربه بوده و اصول و قواعد از پیش تعریف شده‌ای برای آن‌ها وجود ندارد [2]. در این قسمت ما قصد داریم یک نوع معماری از شبکه‌های عصبی را معرفی کنیم. در این نوع معماری، هر مقدار ورودی نرون، با نرون مجاور آن ترکیب می‌شود و خروجی آن ورودی نرون بعدی را معین می‌کند. این نوع معماری را مدل‌های عصبی بازگشتی<sup>۳</sup> می‌نامیم. مدل‌های عصبی بازگشتی یک نوع از شبکه‌های عصبی با معماری خاص است که می‌تواند جهت پردازش داده‌های دارای ساختار و دنباله به ویژه داده‌هایی که توسط گراف جهت‌دار بدون حلقه قابل نمایش است، مورد استفاده قرار بگیرد. معماری‌های شبکه‌ی عمیق شامل چندین لایه‌ی غیرخطی جهت پردازش داده‌ها می‌باشند. به همین علت این نوع شبکه‌ها قادر به ترکیب معنایی داده‌های مربوط به هم و استخراج معنای انتزاعی از آنها می‌باشند. مدل‌های عصبی بازگشتی را می‌توان به عنوان مدلی بسط داده شده از شبکه‌های عصبی Recurrent تلقی نمود [3] که نوع خاصی از ساختار درخت مورب را دارا می‌باشد (همان‌طور در شکل 2-3 می‌توانید مشاهده کنید).

این نوع شبکه‌ها در کاربردهایی نظیر تجزیه<sup>۴</sup> [4]، تحلیل احساسات در سطح جمله [5] [6] و... مورد استفاده قرار گرفته است. با داشتن بازنمایی<sup>۵</sup> ساختارمند یک جمله، مانند درخت تجزیه آن، این نوع شبکه‌ها بازنمایی گره‌های میانی درخت را با توجه به مقدار گره‌های فرزندانش توسط یک رویکرد پایین به بالا، محاسبه

---

<sup>1</sup> Recursive Neural Networks

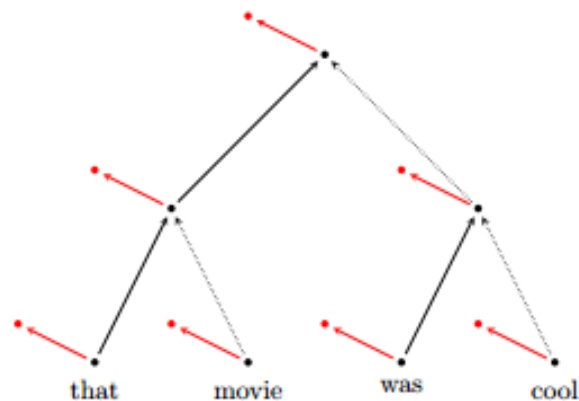
<sup>2</sup> Aproximator

<sup>3</sup> Recursive Neural Models

<sup>4</sup> Parsing

<sup>5</sup> Representation

می‌کنند. با به‌دست آوردن مقدار گره‌های میانی و محاسبه‌ی مقدار گره‌های بالاتر میتوان بازنمایی یک جمله را توسط این رویکرد به‌دست آورد و آنرا جهت انجام کارهایی نظیر طبقه‌بندی<sup>۱</sup> مورد استفاده قرار داد.

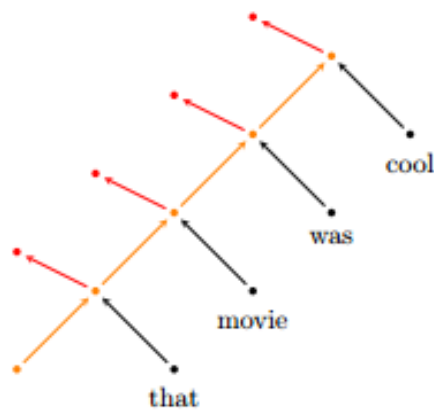


شکل 2-3- ساختار یک شبکه عصبی بازگشتی

معماری مدل‌های عصبی بازگشتی، دارای یک سری وزن‌های مشابه است که به صورت بازگشتی در یک وضعیت دارای ساختار مورد استفاده قرار گرفته می‌شوند: یک گراف بدون دور داده شده، این شبکه گره‌های این گراف را به ترتیب توپولوژیکی بازدید می‌کند و به صورت بازگشتی تغییرات به وجود آورده در نودها را به کار می‌گیرد تا بازنمایی نودهای میانی را از بازنمایی بدست آمده از گره فرزندان، بدست آورد. همانطور که در شکل 3-3 مشاهده می‌کنید، در واقع شبکه‌های عصبی Recurrent، یک شبکه عصبی بازگشتی ساده با ساختاری ویژه است.

<sup>۱</sup> Classification





شکل 2-4- ساختار یک شبکه Recurrent

حتی اگر بتوان مدل‌های عصبی بازگشتی را برای هرگونه درخت جهت‌دار بدون دور به کار برد، ما در اینجا فقط درخت‌های دودویی<sup>1</sup> را جهت به کار بردن در این نوع شبکه‌ها در نظر می‌گیریم [5].

یک درخت دودویی داده شده دارای یک بازنمایی آغازین می‌باشد، مثلاً درخت تجزیه یک جمله با بازنمایی بردار ویژگی کلمات در برگ‌هایش، شبکه عصبی بازگشتی بازنمایی هر نود داخلی  $\eta$  را بصورت زیر محاسبه می‌کند.

$x_{\eta} = f(W_L x_{l(\eta)} + W_R x_{r(\eta)} + b)$	فرمول 2-1
---	-----------

چگونگی محاسبه‌ی گره‌های داخلی در شبکه‌های بازگشتی (تابع ترکیب دو گره فرزند و تولید گره والد)

که  $l(\eta)$  و  $r(\eta)$  فرزندان چپ و راست گرهی  $\eta$ ،  $W_L$  و  $W_R$  وزن‌های ماتریسی هستند که فرزندان چپ و راست را به والدشان متصل می‌کنند و  $b$  بردار Bias می‌باشد.  $W_L$  و  $W_R$  برای همه‌ی گره‌ها ماتریس‌های برابری هستند و تفاوتی بین اینکه  $l(\eta)$  و  $r(\eta)$  برگ درخت است یا اینکه یک گره داخلی قابل نمی‌شوند.

در این تعریف، یک تفسیر مورد توجه قرار می‌گیرد: بازنمایی بردار اولیه‌ی<sup>1</sup> فرزندان و والدها در فضا با اندازه‌ی یکسان انجام می‌شود. در مثال درخت تجزیه‌ی یک جمله، درخت بازگشتی، توسط یک تابع ترکیب، بردار

<sup>1</sup> Binary

بازنمایی چند کلمه را با یکدیگر ترکیب می‌کند تا بتواند بردار بازنمایی یک عبارت شامل چند کلمه را در فضای معنایی یکسان به دست بیاورد [6]. پس از بدست آوردن بردار بازنمایی یک گره، در یک مرحله دیگر، بازنمایی گره را توسط فرمول زیر به خروجی<sup>2</sup> تبدیل می‌کند.

$y_{\eta} = g(U_{x(\eta)} + c)$	فرمول 2-2
---------------------------------	-----------

تبدیل بردار کلمات به برچسب کلاس

که در آن،  $U$  ماتریس وزن خروجی در فضای  $\mathbb{R}^{c \times d}$  ( $c$  تعداد کلاس‌ها و  $d$  طول بردار بازنمایی گره است) و  $C$  بردار Bias می‌باشد. در انجام یک پروژه بانظارت<sup>3</sup>  $y_{\eta}$  خروجی (برچسب کلاس) گره  $\eta$  می‌باشد و نظارت در این لایه انجام می‌گیرد. به عنوان مثال در مسئله‌ی تجزیه و تحلیل احساسات، مقدار  $y_{\eta}$  می‌تواند مقدار برچسب پیش‌بینی شده برای یک عبارت داده شده توسط زیردرخت با ریشه‌ی  $\eta$  باشد. سپس طی روال یادگیری بانظارت، خطای بین مقدار پیش‌بینی شده و مقدار واقعی محاسبه شده و از ریشه درخت به سمت برگ‌ها منتشر می‌شود [14].

تفاوت بین مدل‌های عصبی بازگشتی مختلف در نحوه‌ی محاسبه‌ی بردار بازنمایی والد فرزندان که در فضای  $\mathbb{R}^d$  است، در یک رویکرد پایین به بالا می‌باشد. در ادامه به معرفی مدل‌های عصبی بازگشتی می‌پردازیم.

---

<sup>1</sup> Initial representation

<sup>2</sup> منظور از خروجی برچسبی است که به نمونه تعلق می‌گیرد.

<sup>3</sup> Supervise

### 2.2.1.1 شبکه‌های عصبی بازگشتی<sup>۱</sup>:

ساده‌ترین نوع مدل‌های عصبی بازگشتی، شبکه‌های عصبی بازگشتی می‌باشند که در سال 1996 توسط Goller و همکارانش معرفی گردید و در سال 2011 توسط Socher و همکارانش بسط داده شد [5]. در این مدل، ابتدا مشخص می‌شود که کدام والد تمام گره‌های فرزندش محاسبه شده، به‌عنوان مثال، در درخت داده شده در شکل 2-5،  $p_1$  دارای دو برادر ویژگی فرزندانش (که کلمه هستند) را دارد. شبکه‌های عصبی بازگشتی از فرمول زیر جهت محاسبه‌ی برادر بازنمایی یک والد استفاده می‌کنند:

$$p_1 = f\left(w \begin{pmatrix} b \\ c \end{pmatrix}\right), \quad p_2 = f\left(w \begin{pmatrix} a \\ p_1 \end{pmatrix}\right)$$

که  $f = \tanh$  است و  $w \in \mathbb{R}^{d \times 2d}$  پارامتر اصلی است که باید در طی انجام الگوریتم آموزش داده شود. اگر یک درایه با مقدار 1 به انتهای الحاق<sup>۲</sup> دو برادر فرزندان اضافه<sup>۳</sup> شود، در آن صورت می‌توان بایاس را به عنوان یک ستون اضافی در  $W$  در نظر گرفت.

بردار والد‌ها می‌بایست اندازه‌ی یکسانی با برادر برگ‌ها<sup>۴</sup> داشته باشند تا در ترکیب کردن دو والد با یکدیگر با مشکل تفاوت در ابعاد مواجه نشویم. به همین دلیل ماتریس  $w$  را در ابعاد  $\mathbb{R}^{d \times 2d}$  در نظر می‌گیرند که هنگامی که با ماتریس فرزندان (که به هم الحاق شده‌اند و دارای ابعاد  $\mathbb{R}^{2d \times 1}$  هستند) ضرب می‌شود برداری در ابعاد  $\mathbb{R}^{d \times 1}$  تولید کند.

---

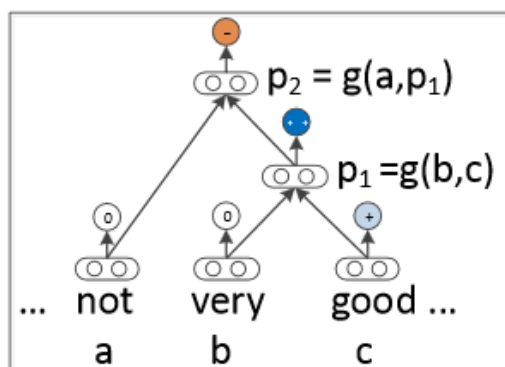
<sup>1</sup> Recursive Neural Network

<sup>2</sup> Concatenation

<sup>3</sup> Append

<sup>4</sup> Leaves

در نهایت هر بردار والد  $p_i$  را به یک طبقه‌بند<sup>1</sup> مانند Softmax داده می‌شود تا احتمال برجسب کلاس‌ها را محاسبه کند.



شکل 2-5- درخت باینری یک عبارت سه کلمه ای<sup>2</sup>

این مدل از تابع ترکیب<sup>3</sup> یکسان جهت رمزنگاری خودکار<sup>4</sup> بازگشتی استفاده می‌کند. در تجربیات عملی ثابت شده است که این مدل با افزایش مقدار داده‌ی آموزش، دقت بیشتری به دست نمی‌آورد [6].

### 2.2.1.2 شبکه‌های عصبی بازگشتی ماتریس-بردار<sup>5</sup>:

در شبکه‌های عصبی بازگشتی، تمام پارامترها و ویژگی‌ها از یک کلمه به دست می‌آیند و هر تابع ترکیبی که ویژگی‌ها را برای عبارات طولانی‌تر محاسبه می‌کند، وابسته به کلمه و ویژگی‌های آن می‌باشد. ایده‌ی اصلی MV-RNN، بازنمایی ویژگی‌های یک کلمه در درخت تجزیه‌ی یک عبارت یا جمله توسط یک بردار و یک ماتریس می‌باشد. هنگام ترکیب دو کلمه و ساخت یک عبارت، ماتریس یک جز با بردار جز دیگر در هم‌دیگر

<sup>1</sup> Classifier

<sup>2</sup> Tri-gram binary tree

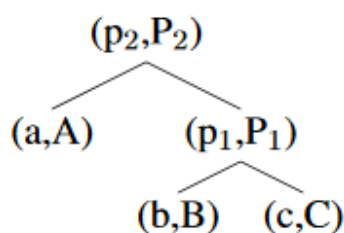
<sup>3</sup> Compositionality function

<sup>4</sup> Auto encoder

<sup>5</sup> MV-RNN: Matrix Vector - RNN

ضرب می‌شود و بالعکس. بنابراین، تابع ترکیب، پارامترهایش را از کلمات یک عبارت ترکیب شده بدست می‌آورد.

ماتریس شناسایی<sup>۱</sup> هر کلمه در ابعاد  $d \times d$  می‌باشد که دارای مقداری نویز گوسین نیز است. پارامترهای این ماتریس آموزش داده خواهند شد تا خطای طبقه‌بندی را در هر گره کمتر کنند. برای این مدل، هر  $n$ -gram توسط یک لیست از جفت ماتریس‌ها و بردارهای ویژگی نمایش داده شده است. برای درخت با گرهی دارای (ماتریس-بردار)، MV-RNN بردار و ماتریس اولین والد توسط فرمول 1-3 محاسبه خواهد کرد:



شکل 2-6- درخت تجزیه‌ی یک عبارت سه کلمه‌ای شامل ماتریس و بردار ویژگی

$p_1 = f\left(w \begin{pmatrix} Cb \\ Bc \end{pmatrix}\right), \quad P_1 = f\left(w_M \begin{pmatrix} B \\ C \end{pmatrix}\right)$	فرمول 1-3
--	-----------

$p_1$  در اینجا نشان‌دهنده‌ی بردار ویژگی و  $P_1$  نشان‌دهنده‌ی ماتریس بحث شده می‌باشد، همچنین  $w_M \in \mathbb{R}^{d \times 2d}$  و پاسخ این معادله در نهایت یک ماتریس  $d \times d$  خواهد بود. به‌طور مشابه، دومین گره والد،

---

<sup>1</sup> Identity

توسط بردار و ماتریس محاسبه شده برای اولین گره والد محاسبه خواهد شد. در انتها، بردارهای به دست آمده با استفاده از فرمول 4 جهت انجام طبقه‌بندی مورد استفاده قرار می‌گیرند.

### 2.3 چالش‌ها

از جمله چالش‌هایی که این پژوهش با آن مواجه است، می‌توان به گردآوری داده جهت آموزش مدل اشاره کرد، با افزایش میزان داده‌ها در فضای مجازی، تصمیم‌گیری در مورد این که چه مقدار داده تهیه شود تا هم خطای مدل کاهش یابد و هم هزینه آماده‌سازی داده کاهش یابد، یکی از چالش‌های انجام این امر می‌باشد. همچنین پس از گردآوری داده‌ها، تصمیم در مورد این که کدام داده، در مجموعه‌ی گردآوری شده باید به عنوان اضافه در مرحله پاک‌سازی داده‌ها، پاک شود از دیگر چالش‌های انجام این تحقیق می‌باشد. چالش سومی که وجود دارد، چگونگی برچسب‌گذاری داده‌ها جهت آموزش مدل می‌باشد. سه چالش مطرح شده، گردآوری، پاک‌سازی و برچسب‌گذاری داده‌ها، در مرحله‌ی تهیه‌ی مجموعه‌داده برای آموزش مدل مطرح می‌باشد. پس از آن، چالشی که در مرحله‌ی پیاده‌سازی مطرح است، انتخاب ابزار مناسبی که هم دسترسی به ابزارهای کاربردی داشته باشد، هم به لحاظ پردازشی سرعت بالایی داشته باشد، می‌باشد.

### 2.4 ابزارهای قابل به کارگیری

جهت پیاده‌سازی مدل، از ابزارهای بسیاری می‌توان کمک گرفت. از جمله این ابزارها می‌توان به نرم‌افزار متلب و زبان برنامه‌نویسی پایتون اشاره نمود. ابزارهای دیگری برای پیاده‌سازی این مدل وجود دارد که به نسبت دو ابزار معرفی شده، کاربرد کمتری دارند، مانند زبان برنامه نویسی جاوا یا C، اما دو ابزار معرفی شده در زمینه پیاده‌سازی اینگونه مدل‌ها در سالهای اخیر به وفور به کار برده شده است. با توجه به اینکه زبان برنامه‌نویسی

پایتون، دارای کتابخانه‌های مختلفی جهت پیاده‌سازی شبکه‌های عصبی عمیق و محاسبات ماتریسی و برداری است، ترجیحاً این زبان به عنوان زبان مورد نظر انتخاب شد.

## 2.5 نتیجه‌گیری

در این فصل مسئله‌ای که پیش‌تر معرفی شده بود، مورد بررسی دقیق‌تر قرار گرفت و همچنین کارهایی که پیش از این برای حل این مسئله انجام شده بودند، معرفی شدند، سپس چالش‌هایی که این مسئله با آنها مواجه است، شرح داده شد. در نهایت ابزارهایی که می‌توانستند متدها و روش‌های معرفی شده را پیاده‌سازی کنند معرفی و یکی از آنها جهت ادامه پژوهش انتخاب شد.

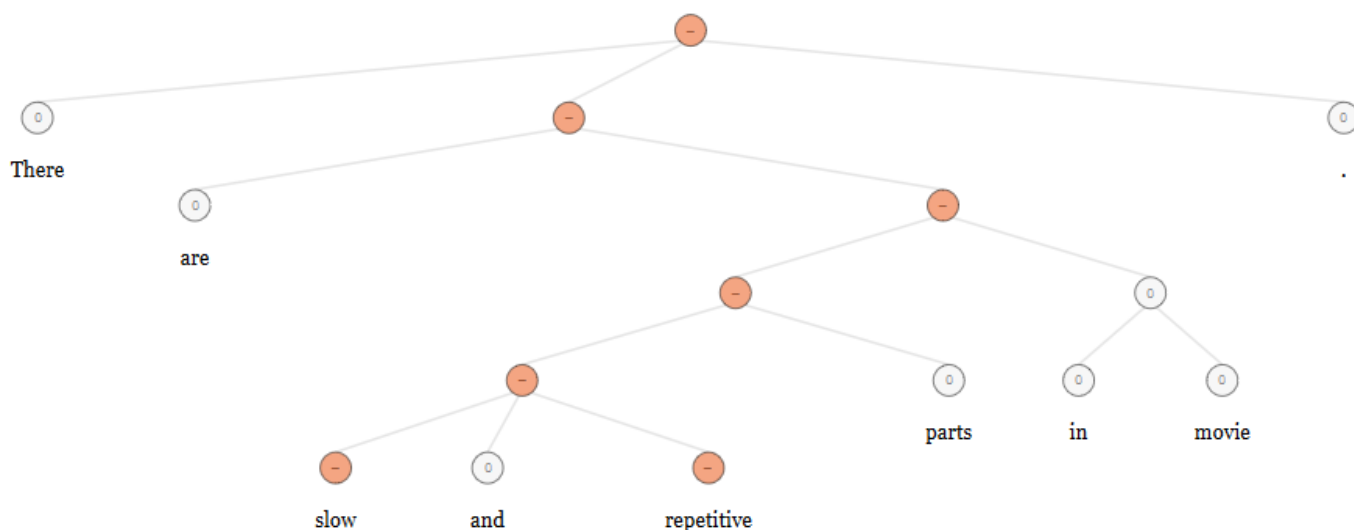




## 3 متدولوژی

### 3.1 فرضیه‌ها

در این فصل، با توجه به فرضیه‌هایی که ارائه خواهد شد، متدولوژی مد نظر معرفی خواهد شد. در این بخش سه فرضیه اساسی مطرح می‌شود؛ فرضیه اول اینست که هر جمله را میتوان به اجزا و ساختار تشکیل دهنده آن تبدیل نمود. برای ایجاد اجزای سازنده‌ی جمله، آنرا به صورت نمایش درختی در می‌آوریم. برای انجام این کار، از ابزار Stanford Parser کمک می‌گیریم. این ابزار توسط Klein و همکارانش در سال 2003 معرفی شده است [7]. منظور از ساختار درختی جمله، تجزیه‌ی جمله به کلمات تشکیل دهنده‌ی آن است. این درخت از نوع دودویی می‌باشد که هر برگ آن یک کلمه از جمله را نشان می‌دهد و هر گره پدر، نشان دهنده‌ی یک عبارت که از ترکیب دو کلمه فرزند آن تشکیل شده است. در شکل 3-1 می‌توانید ساختار تجزیه‌شده‌ی درختی جمله‌ی “There are slow and repetitive parts in movie” را مشاهده کنید.



شکل 3-1- چگونگی نمایش یک جمله بصورت تجزیه شده (درختی)

فرضیه دوم را می‌توان در مورد تغییر کارایی مدل هنگامی که در پارامترهای آن تغییری ایجاد می‌شود، دانست؛ به عبارت دیگر با تغییر در معماری شبکه و پارامترهای ترکیب کلمات در کارایی برچسب‌زنی ماشینی تغییر خواهیم داشت. این فرضیه در تغییر پارامترهای مدل RNN و تبدیل آن به MV-RNN دیده خواهد شد که تغییر مثبت در نتیجه‌ی مدل به‌وجود آورده است. فرضیه سوم را با عنوان کارایی شبکه‌های بازگشتی در برچسب‌زنی کلمات، عبارات و در نهایت جملات دانست. به بیانی دیگر برچسب عبارات حاصل از ترکیب چند کلمه، توسط شبکه‌های عصبی بازگشتی قابل تشخیص است. این فرضیه با در نظر گرفتن اینکه اگر ورودی‌های شبکه اندازه‌ی متغیر نداشته باشند، قابل قبول می‌باشد. پس ترکیب دو کلمه باید به‌گونه‌ای باشد که بازنمایی<sup>1</sup> عبارت تشکیل شده اندازه‌ی یکسانی با بازنمایی یک کلمه داشته باشد تا بتوان مجدداً از آن استفاده کرد و آنرا با ترکیب با یک کلمه‌ی دیگر دوباره وارد شبکه نمود.

به‌عنوان جمع‌بندی، در این بخش سه فرضیه مطرح شد که در زیر آمده است:

<sup>1</sup> Representation

- هر جمله را میتوان به اجزا و ساختار تشکیل دهنده آن تبدیل نمود.
- با تغییر در معماری شبکه و پارامترهای ترکیب کلمات در کارآیی برچسب زنی ماشینی تاثیر دارد.
- برچسب عبارات حاصل از ترکیب چند کلمه، توسط شبکه‌های عصبی بازگشتی قابل تشخیص است.

## 3.2 متدولوژی به کار برده شده

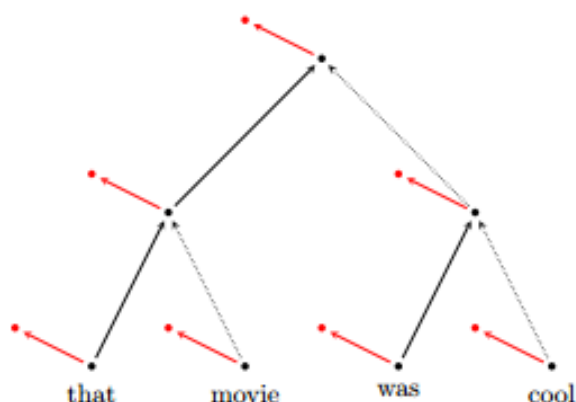
شبکه‌های عصبی، به عنوان یک تخمین‌زننده<sup>1</sup> قوی در بسیاری از زمینه‌ها به کار برده شده است [1]. معماری این شبکه‌ها براساس تجربه بوده و اصول و قواعد از پیش تعریف شده‌ای برای آن‌ها وجود ندارد [2]. در این قسمت ما قصد داریم یک نوع معماری از شبکه‌های عصبی را معرفی کنیم. در این نوع معماری، هر مقدار ورودی نرون، با نرون مجاور آن ترکیب می‌شود و خروجی آن ورودی نرون بعدی را معین می‌کند. این نوع معماری را مدل‌های عصبی بازگشتی<sup>2</sup> می‌نامیم. مدل‌های عصبی بازگشتی یک نوع از شبکه‌های عصبی با معماری خاص است که می‌تواند جهت پردازش داده‌های دارای ساختار و دنباله به ویژه داده‌هایی که توسط گراف جهت‌دار بدون حلقه قابل نمایش است، مورد استفاده قرار بگیرد. معماری‌های شبکه‌ی عمیق شامل چندین لایه‌ی غیرخطی جهت پردازش داده‌ها می‌باشند. به همین علت این نوع شبکه‌ها قادر به ترکیب معنایی داده‌های مربوط به هم و استخراج معنای انتزاعی از آنها می‌باشند. مدل‌های عصبی بازگشتی را می‌توان به عنوان مدلی بسط داده شده از شبکه‌های عصبی Recurrent تلقی نمود [3] که نوع خاصی از ساختار درخت مورب را دارا می‌باشد (همان‌طور در شکل 3-2 می‌توانید مشاهده کنید).

---

<sup>1</sup> Aproximator

<sup>2</sup> Recursive Neural Models

این نوع شبکه‌ها در کاربردهایی نظیر تجزیه<sup>۱</sup> [4]، تحلیل احساسات در سطح جمله [5] [6] و... مورد استفاده قرار گرفته است. با داشتن بازنمایی<sup>۲</sup> ساختارمند یک جمله، مانند درخت تجزیه آن، این نوع شبکه‌ها بازنمایی گره‌های میانی درخت را با توجه به مقدار گره‌های فرزندانش توسط یک رویکرد پایین به بالا، محاسبه می‌کنند. با به‌دست آوردن مقدار گره‌های میانی و محاسبه‌ی مقدار گره‌های بالاتر میتوان بازنمایی یک جمله را توسط این رویکرد به‌دست آورد و آنرا جهت انجام کارهایی نظیر طبقه‌بندی<sup>۳</sup> مورد استفاده قرار داد.



شکل 3-2- ساختار یک شبکه عصبی بازگشتی

معماری مدل‌های عصبی بازگشتی، دارای یک سری وزن‌های مشابه است که به صورت بازگشتی در یک وضعیت دارای ساختار مورد استفاده قرار گرفته می‌شوند: یک گراف بدون دور داده شده، این شبکه گره‌های این گراف را به ترتیب توپولوژیکی بازدید می‌کند و به صورت بازگشتی تغییرات به وجود آورده در نودها را به‌کار می‌گیرد تا بازنمایی نودهای میانی را از بازنمایی بدست آمده از گره فرزندان، بدست آورد. همانطور که در

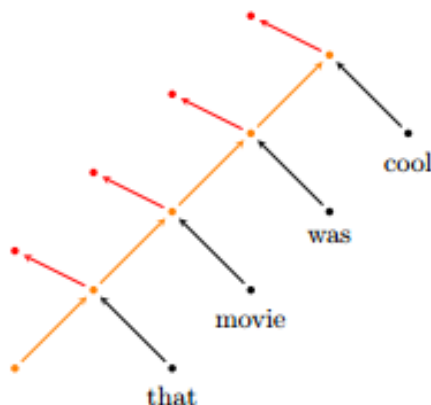
---

<sup>1</sup> Parsing

<sup>2</sup> Representation

<sup>3</sup> Classification

شکل 3-3 مشاهده می کنید، در واقع شبکه های عصبی Recurrent، یک شبکه عصبی بازگشتی ساده با ساختاری ویژه است.



شکل 3-3- ساختار یک شبکه Recurrent

حتی اگر بتوان مدل های عصبی بازگشتی را برای هرگونه درخت جهت دار بدون دور به کار برد، ما در اینجا فقط درخت های دودویی<sup>1</sup> را جهت به کار بردن در این نوع شبکه ها در نظر می گیریم [5].

یک درخت دودویی داده شده دارای یک بازنمایی آغازین می باشد، مثلاً درخت تجزیه یک جمله با بازنمایی بردار ویژگی کلمات در برگهایش، شبکه عصبی بازگشتی بازنمایی هر نود داخلی  $\eta$  را بصورت زیر محاسبه می کند.

فرمول 2-3	$x_{\eta} = f(W_L x_{l(\eta)} + W_R x_{r(\eta)} + b)$
-----------	---

چگونگی محاسبه ی گره های داخلی در شبکه های بازگشتی (تابع ترکیب دو گره فرزند و تولید گره والد)

که  $l(\eta)$  و  $r(\eta)$  فرزندان چپ و راست گره ی  $\eta$ ،  $W_L$  و  $W_R$  وزن های ماتریسی هستند که فرزندان چپ و راست را به والدشان متصل می کنند و  $b$  بردار Bias می باشد.  $W_L$  و  $W_R$  برای همه ی گره ها ماتریس های برابری هستند و تفاوتی بین اینکه  $l(\eta)$  و  $r(\eta)$  برگ درخت است یا اینکه یک گره داخلی قایل نمی شوند.

<sup>1</sup> Binary

در این تعریف، یک تفسیر مورد توجه قرار می‌گیرد: بازنمایی بردار اولیه<sup>۱</sup> فرزندان و والدها در فضا با اندازه‌ی یکسان انجام می‌شود. در مثال درخت تجزیه‌ی یک جمله، درخت بازگشتی، توسط یک تابع ترکیب، بردار بازنمایی چند کلمه را با یکدیگر ترکیب می‌کند تا بتواند بردار بازنمایی یک عبارت شامل چند کلمه را در فضای معنایی یکسان به‌دست بیاورد [6]. پس از بدست آوردن بردار بازنمایی یک گره، در یک مرحله دیگر، بازنمایی گره را توسط فرمول زیر به خروجی<sup>۲</sup> تبدیل می‌کند.

$y_{\eta} = g(U_{x(\eta)} + c)$	فرمول 2-4
---------------------------------	-----------

تبدیل بردار کلمات به برجسب کلاس

که در آن،  $U$  ماتریس وزن خروجی در فضای  $\mathbb{R}^{c \times d}$  ( $c$  تعداد کلاس‌ها و  $d$  طول بردار بازنمایی گره است) و  $C$  بردار Bias می‌باشد. در انجام یک پروژه بانظارت<sup>۳</sup>  $y_{\eta}$  خروجی (برجسب کلاس) گره  $\eta$  می‌باشد و نظارت در این لایه انجام می‌گیرد. به‌عنوان مثال در مسئله‌ی تجزیه و تحلیل احساسات، مقدار  $y_{\eta}$  می‌تواند مقدار برجسب پیش‌بینی شده برای یک عبارت داده شده توسط زیردرخت با ریشه‌ی  $\eta$  باشد. سپس طی روال یادگیری بانظارت، خطای بین مقدار پیش‌بینی شده و مقدار واقعی محاسبه شده و از ریشه درخت به سمت برگ‌ها منتشر می‌شود [14].

تفاوت بین مدل‌های عصبی بازگشتی مختلف در نحوه‌ی محاسبه‌ی بردار بازنمایی والد فرزندان که در فضای  $\mathbb{R}^d$  است، در یک رویکرد پایین به بالا می‌باشد. در ادامه به معرفی مدل‌های عصبی بازگشتی می‌پردازیم.

<sup>1</sup> Initial representation

<sup>2</sup> منظور از خروجی برجسبی است که به نمونه تعلق می‌گیرد.

<sup>3</sup> Supervise

### 3.2.1 شبکه تنسور<sup>۱</sup> عصبی بازگشتی<sup>۲</sup> (RNTN) :

یکی از مشکلات شبکه‌های عصبی بازگشتی ماتریس-بردار این است که تعداد پارامترهای مدل براساس اندازه‌ی واژگان افزایش می‌یابد. داشتن تابعی که بتواند ویژگی‌های فرزندان را با تعداد پارامترهای ثابت ترکیب کند بسیار مطلوب و مورد پسند می‌باشد. تغییر دادن شبکه‌های RNN ایده‌ی بسیار خوبی جهت پیاده‌سازی چنین تابع ترکیبی است. در شبکه‌های استاندارد RNN، بردارهای ویژگی وارد شده به شبکه، به‌صورت مستقیم توسط توابع غیرخطی با یکدیگر ارتباط دارند. به‌طور کلی به نظر می‌رسد تعامل بیشتر بین بردارهای ورودی باعث افزایش دقت مدل خواهد شد.

بر اساس ایده‌های مطرح شده، چند سوال پیش می‌آید: آیا می‌توان یک تابع ترکیب قدرتمندی ایجاد کرد که بهتر از توابع قبلی عمل کند و اجزای کوچکتر یک عبارت را به صورت معنادارتری با یکدیگر ترکیب کند؟ جهت جواب دادن به این سوالات، در این بخش مدلی به نام شبکه تنسور عصبی بازگشتی معرفی می‌شود. ایده‌ی اصلی، استفاده از یک تابع ترکیب مبتنی بر تنسور برای تمام گره‌ها می‌باشد.

در شکل 3-4 می‌توانید یک لایه‌ی تنسوری را مشاهده کنید. خروجی ضرب تنسور،  $h \in \mathbb{R}^d$  توسط معادله‌ی زیر معین می‌شود. اما نشان‌گذاری جزئی‌تر هر تکه<sup>۳</sup> به صورت  $v^{[i]} \in \mathbb{R}^{d \times d}$  می‌باشد.

$$h = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix}; h_i = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[i]} \begin{bmatrix} b \\ c \end{bmatrix}.$$

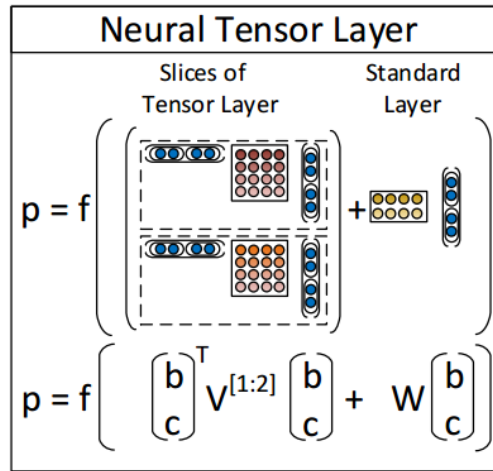
که  $v^{[1:d]} \in \mathbb{R}^{2d \times 2d \times d}$  تنسوری است که چندین فرم دوطرفه خطی را تعیین می‌کند.

---

<sup>1</sup> Tensor

<sup>2</sup> Recursive Neural Tensor Network

<sup>3</sup> Slice



شکل 3-4- یک لایه از شبکه تنسور عصبی بازگشتی، قسمت‌های خط‌چین یک تکه از لایه‌ی تنسوری را نمایش می‌دهد

تابع ترکیب RNTN از معادله‌ی زیر جهت محاسبه‌ی  $p_1$  استفاده می‌کند.

$$p_1 = f \left( \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix} \right),$$

که  $w$  در مدل قبلی تشریح شد. بردار والد بعدی، در یک عبارت سه کلمه‌ای، با همین وزن‌ها محاسبه خواهد شد.

مزیت اصلی این مدل نسبت به مدل RNN، که اگر  $v$  را برابر صفر در نظر بگیریم، مدل RNN، را می‌توان نوع خاصی از مدل RNTN تلقی نمود، این است که تنسور به‌طور مستقیم می‌تواند به بردارهای ورودی مرتبط باشد.

$$p_2 = f \left( \begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} \right).$$

در کل، می‌توان هر تکه از تنسور را به‌عنوان نشان‌دهنده‌ی نوع خاصی از ترکیب تفسیر نمود.



اضافه نمودن یک لایه شبکه عصبی دیگر، می‌تواند برای RNTN تابع ترکیب قوی‌تری را ایجاد کند. اما تجربیات اولیه نشان داده‌اند که بهینه‌سازی این مدل سخت و پیچیده است و تعاملات بردارها ضمنی‌تر<sup>۱</sup> از RNTN است.

### 3.3 جمع بندی

در این فصل، الگوریتم‌های مبتنی بر شبکه‌های عصبی بازگشتی، که در طبقه‌بندی متن کاربرد داشتند، معرفی شدند. فرمول‌ها، روابط و پارامترهایی که باید در طول آموزش، تغییر کنند بطور مفصل توضیح داده شد و سه مدل از بازگشتی‌ها معرفی شدند. این سه مدل عبارت بودند از RNN، MV-RNN و RNTN که هر یک به ترتیب کامل شده یا تغییر یافته‌ی الگوریتم قبلی بود. در این میان، RNTN رویکردی است که در این پایان‌نامه قصد داریم آنرا با دیگر الگوریتم‌های معرفی شده مقایسه کنیم. در فصل بعدی، نتایج عملی حاصل از پیاده‌سازی این الگوریتم‌ها ارائه خواهد شد.

---

<sup>1</sup> More implicit



## 4 پیاده‌سازی و ارزیابی

### 4.1 مجموعه داده‌ها

در این پروژه برای آموزش و آزمون مدل از مجموعه داده‌های بانک درختی نظرات استنفورد<sup>۱</sup> استفاده شد. در ادامه این بخش مراحل و چگونگی گردآوری<sup>۲</sup>، پاک‌سازی<sup>۳</sup> و برچسب‌گذاری<sup>۴</sup> جملات، عبارات و کلمات تشریح خواهد شد.

مجموعه داده‌های بانک درختی نظرات استنفورد شامل 10662 جمله است که به‌طور تقریبی نیمی از آنها را جملات مثبت و نیمی دیگر از آنها را جملات منفی تشکیل می‌دهند. این مجموعه داده توسط مقاله [8] جمع‌آوری و انتشار یافته است.

پس از برداشت جملات از سایت، ابتدا همه حروف به حروف کوچک انگلیسی تبدیل شد؛ سپس زواید و اضافات موجود در جملات که شامل شکلک‌ها<sup>۵</sup>، برچسب‌های<sup>۶</sup> HTML و عباراتی که در زبان انگلیسی موجود نبودند، حذف شدند.

---

<sup>1</sup> Stanford sentiment treebank

<sup>2</sup> Gathering

<sup>3</sup> Cleaning

<sup>4</sup> Labeling

<sup>5</sup> Emoji

<sup>6</sup> Tags

در گام سوم ابزار تجزیه‌ی جملات به درخت تشکیل‌دهنده‌ی آن، که توسط Klein و همکارانش در سال 2003 در دانشگاه استنفورد و در مقاله‌ی [7] معرفی شد، مورد استفاده قرار گرفت تا تمام 10662 جمله به درخت تشکیل‌دهنده‌ی آن تبدیل شود.

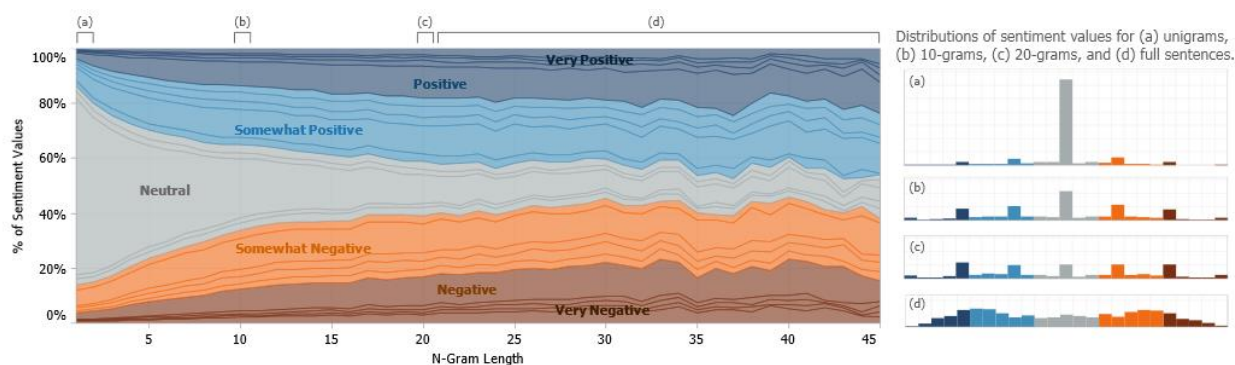
پس از انجام مراحل فوق، در گام چهارم، تمام 215154 عبارات و کلمات به‌دست آمده از درخت جملات توسط ابزار مکانیکال‌تورک آمازون<sup>1</sup> برچسب‌گذاری شدند. مکانیکال‌تورک یک ابزار جهت جمع‌سپاری کارهای بزرگ می‌باشد. شکل 1-4 ابزار رابط برچسب‌گذاران را نمایش می‌دهند. در این مرحله عبارات و جملات به‌صورت تصادفی به برچسب‌گذاران نمایش داده می‌شود و آنها با استفاده از این ابزار قطبیت جمله را معین می‌کنند.



شکل 1-4- رابط برچسب‌گذاری. عبارات بصورت تصادفی به برچسب‌گذاران نمایش داده می‌شد و آنها درجه‌ی مثبت یا منفی بودن آنها را مشخص می‌کنند [5].

رابط کاربری دارای 25 مقدار مختلف است که به‌طور پیش‌فرض بر روی قطبیت خنثی تنظیم شده است. شکل 2-4 نمودار هیستوگرام نرمال شده‌ی برچسب‌های عبارات را نشان می‌دهد. همان‌طور که مشاهده می‌کنید، بسیاری از کلمات قطبیت خنثی دارند، عبارات بیش از یک کلمه، به‌صورت مطلوبی پخش شده‌اند و توزیع یکنواخت دارند.

<sup>1</sup>Amazon Mechanical Turk



شکل 4-2- نمودار هیستوگرام نرمال شده‌ی برچسب‌های عبارات

یکنواخت بودن توزیع عبارات بسیار مهم است زیرا به عنوان مثال، اگر نود درصد عبارات و کلمات قطبیت خنثی داشته باشند، یک طبقه‌بند، تمام عبارات را خنثی در نظر می‌گیرد و دقت آن برابر نود درصد می‌باشد و این مدل بر روی عبارات و کلمات خارج از مجموعه داده پاسخ با خطای زیاد برمی‌گرداند.

## 4.2 ارزیابی نمودارهای حاصل شده

پس از پیاده‌سازی، مدل به دو صورت مورد تحلیل و بررسی قرار گرفت. تحلیل و بررسی اول، شامل چندین معیار اعتبارسنجی کیفیت مدل بر روی داده‌های تست است. نوع دوم تحلیل مدل، به بررسی دو پدیده‌ی زبان‌شناسی که در نظرات مهم هستند می‌پردازد.

برای تمامی مدل‌ها، از اعتبارسنجی موازی<sup>۱</sup> برای تنظیم مقادیر ماتریس وزن‌ها، بردار کلمه، نرخ یادگیری<sup>۲</sup> و کمینه‌اندازه‌ی دسته<sup>۳</sup> برای *AdaGrad* مورد استفاده قرار گرفت. دقت بهینه برای تمامی مدل‌ها، در اندازه‌ی

<sup>1</sup> Cross-validation

<sup>2</sup> Learning rate

<sup>3</sup> Minibatch size

25 تا 35 برای بردار کلمات، و اندازه‌ی دسته<sup>۱</sup> بین 20 و 30، به‌دست آمد. اندازه‌ی بردار کلمات یا اندازه‌ی دسته کوچکتر باعث کاهش دقت مدل می‌شد. این جمله به این معنا است که هنگامی که پارامترهای مدل افزایش می‌یابد، مدل *RNTN* بهتر از مدل *RNN* عمل می‌کند.

مدل *MV-RNN* بیشترین تعداد پارامترها در ماتریس کلمات را در بین مدل‌های معرفی شده دارد همچنین مدل *RNTN* پس از حدوداً 60 دوره<sup>۲</sup>، به بیشترین دقت بین مدل‌های معرفی شده دست پیدا می‌کند. تجربیات نشان می‌دهد که مدل‌های بازگشتی، هنگامی که از تابع فعال‌ساز غیرخطی استفاده می‌کند، به طور چشمگیری (بطور متوسط نزدیک به 5٪ کاهش دقت) بد کار می‌کند. در اینجا ما از تابع  $f=tanh$  در تمامی کارها استفاده کردیم.

مدل‌های معرفی شده با مدل‌های *Naïve Bayes* و *SVM* که عموماً از ویژگی‌های کیسه‌کلمات<sup>۳</sup> یک متن برای تحلیل آن استفاده می‌کنند، و یک مدل *SVM* که از ویژگی کلمات به‌صورت دودویی استفاده می‌کرد، مقایسه شد. همین‌طور مدل‌ها را یک مدل که میانگین بردار کلمات را محاسبه می‌کرد (*VecAvg*)، مقایسه شد. این مدل‌ها در جدول مقایسه با عناوین اختصاری NB, SVM و BiSVM بیان شده است. جملات مجموعه‌داده، به سه بخش (*Train*(8544)، *Test*(2210) و *Dev*(1101) جداسازی شد. همچنین کارایی مدل در تشخیص جملاتی که فقط مثبت یا منفی بودند نیز مورد بررسی قرار گرفت. در این حالت از جملاتی که قطبیت خنثی داشتند صرف نظر شد. پس از حذف جملات با قطبیت خنثی، نزدیک به 20٪ از کل داده‌ها حذف شد و سه بخش داده‌های مذکور به ترتیب، 6920/1821/872 تعداد جمله در آن‌ها باقی ماند.

---

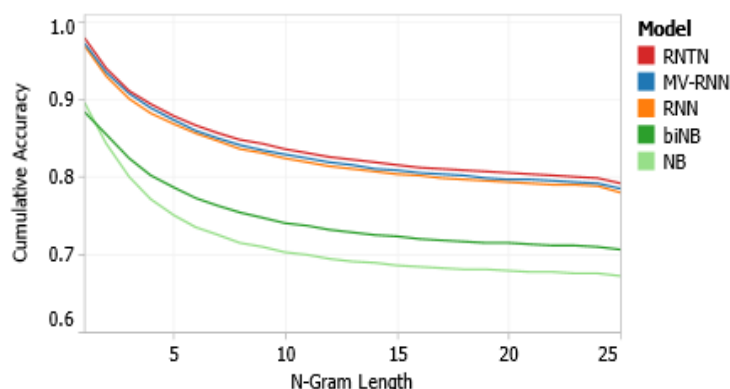
<sup>1</sup> Batch

<sup>2</sup> Epoch

<sup>3</sup> Bag of word

## ریزطبقه‌بندی<sup>۱</sup> عبارات:

معیار اصلی ارزیابی ریزطبقه‌بندی نظرات، دقت مدل در طبقه‌بندی تمام عبارات مجموعه داده می‌باشد. همان‌طور که در بخش مجموعه داده‌ها بیان شد، توزیع جملات در مجموعه داده‌ها به گونه‌ای است که می‌توان آنرا به 5 کلاس با توزیع داده‌های یکنواخت تقسیم کرد. شکل 3-4 نتایج به دست آمده از مدل‌های مختلف را نشان می‌دهد که در مقاله [6] ارائه شده است.

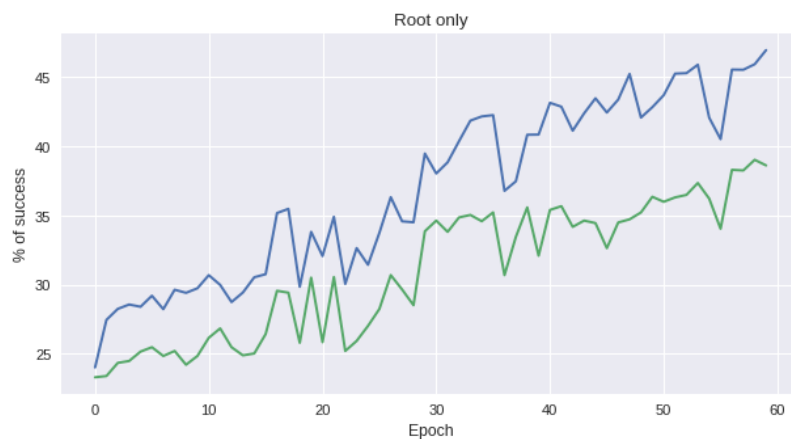


شکل 3-4- نمودار دقت ریزطبقه‌بندی عبارات [6]

همان‌گونه که در شکل 3-4 مشاهده می‌کنید، مدل *RNTN* توانسته بیشترین دقت را در مقایسه با دیگر مدل‌ها به دست آورد. به‌طور خاص نموداری که پس از پیاده‌سازی مدل *RNTN* برای طبقه‌بندی جملات بدست آمد، در شکل 4-4 نمایش داده شده است.

---

<sup>1</sup> Fine-grained classification



شکل 4-4- نمودار به دست آمده از پیاده سازی مدل RNTN جهت طبقه بندی جملات

همچنین نمودار این مدل در طبقه بندی کلمات و عبارات به صورت شکل 5-4 حاصل شد.

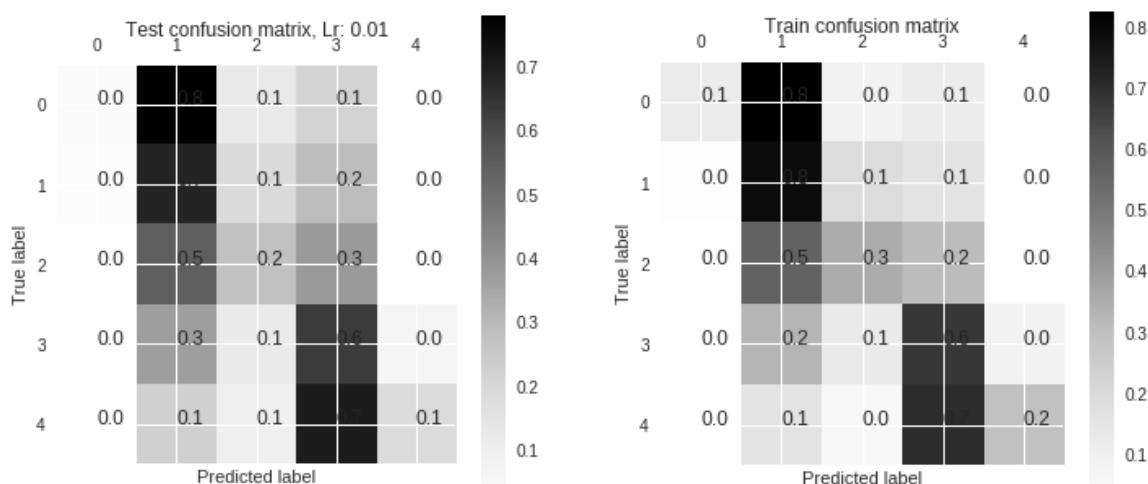


شکل 5-4- نمودار به دست آمده از پیاده سازی مدل RNTN جهت طبقه بندی کلمات

همچنین نمودار ماتریسی سرگشتگی<sup>۱</sup> که پس از وارد کردن داده های آزمون و آموزش به مدل به دست آمد در شکل 6-4 به نمایش در آمده است.

<sup>۱</sup> Confusion Matrix Plot





شکل 4-6 نمودارهای ماتریس سرگشتگی مدل RNTN جهت طبقه‌بندی جملات

در جدول جدول 1-4 می‌توان میزان دقت طبقه‌بندی‌های مختلف را که در مقاله [6] آمده است را مشاهده نمود، همانطور که نمودار دارد، مدل RNTN توانسته است در بین این مدل‌ها بیشترین دقت را به دست آورد. و پس از آن به ترتیب مدل‌های MV-RNN و RNN بیشترین دقت‌ها را به دست آوردند. مدل‌های بازگشتی، دقت بسیار خوبی بر روی عبارات کوتاه‌تر دارند زیرا کلمات منفی و ترکیب آن‌ها مهم است در حالی که مدل‌های مبتنی بر کیسه کلمات<sup>۱</sup> ترکیب کلمات منفی را در نظر نمی‌گیرد.

<sup>۱</sup> Bag-of-word

جدول 4-1 - دقت پیش‌بینی مدل‌ها در حالت ریز طبقه‌بندی (5 طبقه) در دو سطح گره و جمله

Model	5 Class Classification	
	All nodes	Root only
NB	67.2	41.0
SVM	64.3	40.7
BiNB	71.0	41.9
VecAvg	73.3	32.7
RNN	79.0	43.2
MV-RNN	78.7	44.4
<b>RNTN</b>	<b>80.7</b>	<b>45.7</b>

در جدول جدول 4-1 مشاهده می‌شود که بیشترین ضعف مدل RNTN در طبقه‌بندی کردن جملات مجموعه داده می‌باشد.

#### طبقه‌بندی دودویی :

این مدل از طبقه‌بندی قابل مقایسه با مدل [9] است که از مجموعه داده مشابه اما بدون ساختار درختی و برچسب جملات برای طبقه‌بندی جملات به دو کلاس مثبت و منفی استفاده کرده است. با طبقه‌بندی جملات به دو کلاس مثبت و منفی، با ساختار درختی جملات شاهد افزایش دقت پیش‌بینی هستیم. **Error!** **Reference source not found.** نتایج طبقه‌بندی دودویی برای هر دو حالت پیش‌بینی عبارت و پیش‌بینی جمله را نشان می‌دهد. بیشترین دقت به دست آمده پیش از ارائه مدل *RNTN*، 80٪ بوده که توسط [9] به دست آمده است. روش‌های مبتنی بر کیسه کلمات، بسیاری از پدیده‌های زبان‌شناسی را نمی‌توانند مدل‌سازی کند، حتی اگر مدل‌ها بسیار پیچیده‌تر می‌شدند. ترکیب معنایی درخت مفهومی جمله و روش RNTN توانست بیشترین دقت به دست آمده را به 85٪ برساند.

اگر به نمودار ماتریس سرگشتگی که در شکل 4-6 آمده است دقت کنیم، متوجه می‌شویم که مشکل اساسی مدل در شناسایی جملات خیلی مثبت، خیلی منفی و خنثی می‌باشد. با تقلیل تعداد کلاس‌ها از 5 به 2 مدل و در نظر گرفتن طبقه خنثی به عنوان طبقه‌ی مثبت، RNTN توانست بهبود در تشخیص سه کلاس ذکر شده ایجاد کند. در

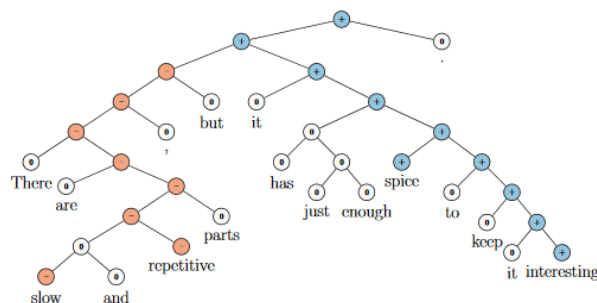
جدول 4-2 خروجی حاصل از تغییرات این مدل آورده شده است.

جدول 4-2 - دقت پیش‌بینی مدل‌ها در حالت طبقه‌بندی‌باینری (2 طبقه) در دو سطح گره و جمله

Model	Binary Classification	
	All nodes	Root only
NB	82.6	81.8
SVM	84.6	79.4
BiNB	82.7	83.1
VecAvg	85.1	80.1
RNN	86.1	82.4
MV-RNN	86.8	82.9
<b>RNTN</b>	<b>87.6</b>	<b>85.4</b>

### تحلیل مدل با جملات دارای حرف ربط عاطفی:

در این بخش، از مجموعه‌ای از جملات که دارای ساختار "... اما ...": یک عبارت چند کلمه‌ای + اما " یک عبارت چند کلمه‌ای، جهت انجام آزمایشات روی مدل، استفاده شد. شکل 4-7 یک مثال از چنین جمله‌ای را نشان می‌دهد. اگر پیش‌بینی هر دو عبارت در دو طرف جمله صحیح باشد، گره ریشه مثبت محاسبه خواهد شد. در نتیجه‌ی 131 جمله، RNTN به دقت 41٪ در تشخیص این‌گونه جملات رسید. در نهایت RNTN به بیشترین مقدار دقت در مقایسه با دیگر مدل‌ها مانند MV-RNN(37)، RNN(36) و BiNB(27) رسید [6].



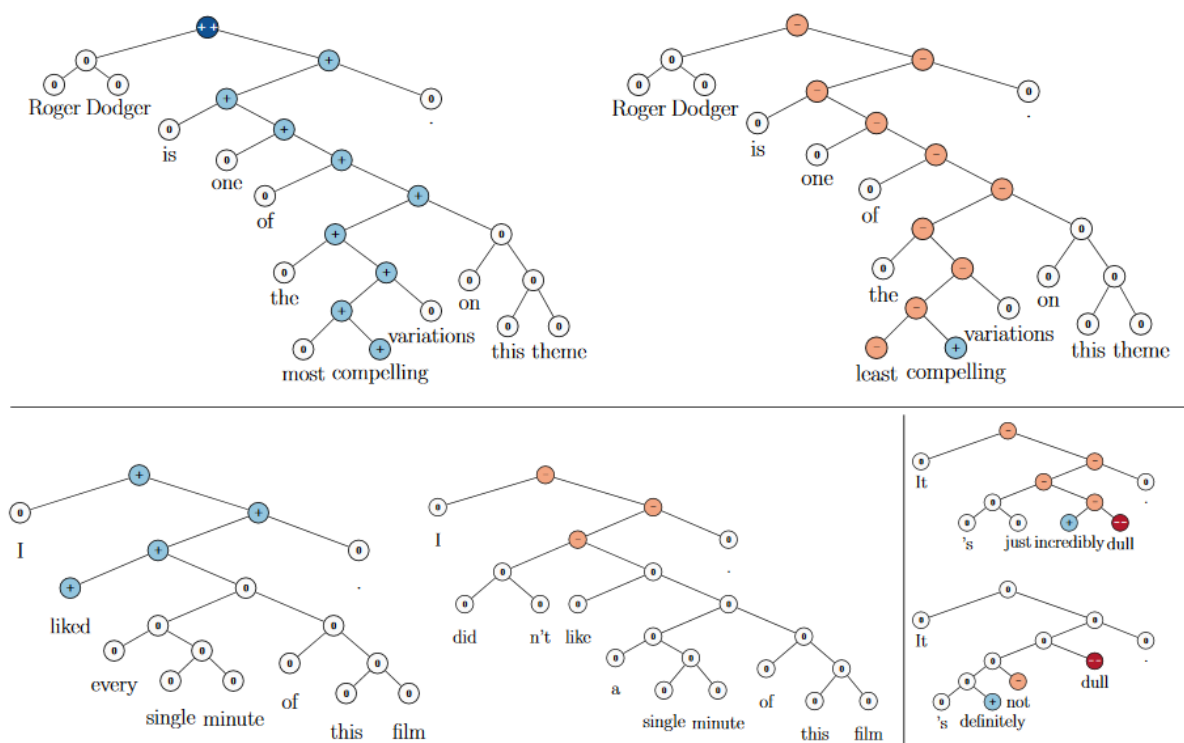
شکل 4-7- یک مثال از پیش‌بینی صحیح مدل در جمله با ساختار ".... اما ..."

**تحلیل مدل با جملات دارای قطبیت منفی سطح بالا:**

در این قسمت مدل با دو نوع از جملات منفی مورد آزمایش قرار داده شد. و برای هر نوع از جملات، مجموعه داده‌ای مجزا جهت تست مدل در نظر گرفته شد.

مجموعه داده اول: معکوس سازی قطبیت جملات مثبت:

مجموعه داده‌ی اول شامل جملات مثبت و جملات منفی شده‌ی آنها بود. در این مجموعه، کلمه‌ی منفی‌ساز، قطبیت تمام جمله را از مثبت به منفی تغییر می‌داد. از این‌رو، دقت مدل بنابر جملات صحیح پیش‌بینی شده توسط مدل، محاسبه می‌شود. شکل 4-8 دو نمونه از قطبیت مثبت معکوس شده را نشان می‌دهد که به‌درستی، توسط مدل دسته‌بندی شده‌اند.



شکل 4-8- پیش‌بینی مدل قطبیت جملات مثبت و منفی و قطبیت معکوس شده‌ی آنها

در جدول 3-4 دقت مدل‌ها بر روی جملات با قطبیت مثبت و جمله‌ی معکوس شده‌ی آنها را مشاهده می‌کنید. در این آزمایش هم مدل RNTN توانست به بیشترین دقت دست یابد.

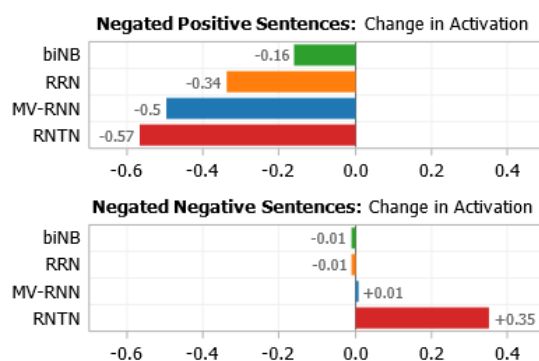
جدول 3-4- دقت مدل در تشخیص جملاتی که قطبیت آنها توسط یک کلمه معکوس شده است

Model	Accuracy	
	Negated Positive	Nagated Negative
biNB	19.0	27.3
RNN	33.3	54.5
MV-RNN	52.4	54.6
<b>RNTN</b>	<b>71.4</b>	<b>81.8</b>

#### مجموعه داده دوم: معکوس سازی قطبیت جملات منفی:

مجموعه داده دوم شامل جملاتی با قطبیت منفی و جملات مثبت شده‌ی آنها است. هنگامی که قطبیت جملات منفی معکوس شد، متوجه شدیم که قطبیت تمام جملات می‌توانند به اندکی منفی تغییر پیدا کنند نه لزوماً مثبت. به عنوان مثال جمله‌ی "فیلم خسته کننده بود" منفی است اما جمله‌ی "فیلم خسته کننده نبود" نشان می‌دهد که نویسنده فیلم را نپسندیده اما نظر خیلی منفی روی فیلم هم ندارد. بنابراین، دقت مدل را بر اساس اینکه چگونه مدل قادر بود جملات غیرمنفی را تشخیص دهد ارزیابی شد. در سمت راست جدول 3-4 دقت مدل بر اساس این مجموعه داده نشان داده شده است. در 81٪ از موارد، RNTN توانسته است پیش‌بینی صحیح از جمله داشته باشد. در قسمت پایین سمت راست شکل 3-4 یک مورد نشان داده شده است که با وجود کلمات منفی Not و Dull، چگونه جمله توسط یک کلمه از قطبیت منفی به قطبیت خنثی تغییر پیدا کرده است. شکل 4-9 تغییرات فعالیت برای هر دو مجموعه داده را نشان می‌دهد، مقادیر منفی شده، باعث افزایش میانگین فعال شدن برای مجموعه داده اول، و افزایش میانگین فعال سازی مقادیر مثبت در مجموعه داده دوم را نشان می‌دهد. همان‌طوری که مشهود است، مدل RNTN بیشترین تغییر را در

جهت درست داشته است. بنابراین می‌توانیم نتیجه بگیریم که مدل RNTN بهترین مدل جهت شناسایی کلمات معکوس کننده قطبیت جمله می‌باشد [10].



شکل 4-9- تغییر فعالیت مدل‌ها با جملاتی که قطبیت آنها معکوس شده

### تحلیل مدل با منفی‌ترین و مثبت‌ترین کلمات:

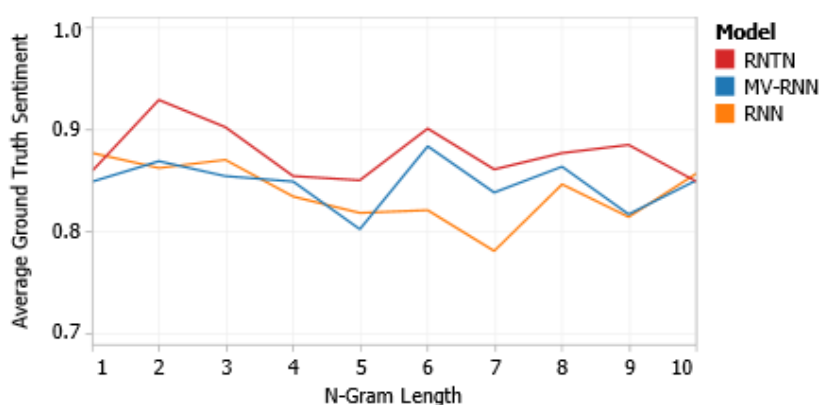
در این قسمت، کدی جهت یافتن مثبت‌ترین و منفی‌ترین عبارات موجود در دایره‌واژگان<sup>1</sup> که مدل شناسایی نموده به مدل اضافه شد. این کد کلماتی را که مدل در مجموعه داده به عنوان کلمات بسیار مثبت و بسیار منفی تشخیص داده بود را نمایش می‌داد. جدول 4-4 شامل مثبت‌ترین و منفی‌ترین کلماتی است که مدل در مجموعه داده شناسایی کرده است.

<sup>1</sup> Vocabulary

جدول 4-4- مثالی از مثبت‌ترین و منفی‌ترین عباراتی که مدل شناسایی کرده است [6]

$n$	Most positive $n$ -grams	Most negative $n$ -grams
1	engaging; best; powerful; love; beautiful	bad; <b>dull</b> ; boring; fails; worst; stupid; painfully
2	excellent performances; A masterpiece; masterful	worst movie; very bad; shapeless mess; worst
3	film; wonderful movie; marvelous performances	thing; instantly forgettable; complete failure
5	an amazing performance; wonderful all-ages triumph; a wonderful movie; most visually stunning	for worst movie; A lousy movie; a complete failure; most painfully marginal; very bad sign
8	nicely acted and beautifully shot; gorgeous imagery, effective performances; the best of the year; a terrific American sports movie; refreshingly honest and ultimately touching	silliest and most incoherent movie; completely crass and forgettable movie; just another bad movie. A cumbersome and cliché-ridden movie; a humorless, disjointed mess
8	one of the best films of the year; A love for films shines through each frame; created a masterful piece of artistry right here; A masterful film from a master filmmaker,	A trashy, exploitative, thoroughly unpleasant experience ; this sloppy drama is an empty vessel.; quickly drags on becoming boring and predictable.; be the worst special-effects creation of the year

شکل 4-10 نشان می‌دهد که میزان دقت مدل RNTN، در عبارات و کلمات با طول‌های مختلف چگونه است [10].



شکل 4-10- میانگین تشخیص مثبت‌ترین و منفی‌ترین عبارات با طول‌های مختلف [10]



### 4.3 توصیه‌ها و کارهای قابل انجام در آینده

#### 4.3.1 پیاده‌سازی در زبان فارسی

بدیهی است که مجموعه داده‌ای که برای آموزش و آزمون مدل استفاده شد، به زبان انگلیسی بود و اینکه آیا می‌توان این چنین مدل‌هایی را در زبان فارسی پیاده‌سازی کرد یا خیر جای بحث و گفت‌وگو دارد. تهیه‌ی مجموعه داده‌ی فارسی طبق فرمتی که در بخش 4.1 معرفی شد و سپس آموزش و آزمون آن با مدل‌های معرفی شده می‌تواند جز کارهایی باشد که در آینده می‌تواند انجام بگیرد. مشکلی که در انجام این کار می‌تواند پیش بیاید، هزینه بر بودن تهیه‌ی این چنین مجموعه داده‌ای می‌باشد. هزینه‌ی جمع‌آوری جملات و پاک‌سازی جملات از موارد زائد و اضافی ناچیز است و می‌تواند توسط پیاده‌ساز با صرف زمان اندکی انجام شود. همچنین تبدیل جملات به ساختار درختی تشکیل دهنده‌ی آن و به دست آوردن کلمات و عبارات، به شرطی که کلمات در غالب صحیح و غیر محاوره‌ای به کار گرفته شده باشد، نیز زمان و هزینه‌ی آنچنانی به کل کار تحمیل نمی‌کند چرا که تجزیه‌کننده جملات فارسی به اجزای تشکیل دهنده‌ی آن وجود دارد. اما اگر کلمات و عبارات محاوره‌ای در جملات وجود داشته باشد، تجزیه‌کننده‌ی این جملات که بتواند جمله را به صورت ساختار درختی تبدیل کند وجود ندارد، و جهت انجام این کار می‌بایست از ابتدا یک تجزیه‌کننده ساخته شود و سپس ادامه کار طبق روندی که در فصل‌های پیش توضیح داده شده انجام گیرد.

آنچه بیش از همه در این روند زمان و هزینه‌بر است، برچسب‌گذاری تک‌تک کلمات، عبارات و جملات است. در صورت آماده‌سازی چنین مجموعه داده‌ای، وارد کردن آن به شبکه عصبی و دیدن نتیجه‌ی آن، کار آنچنان سخت و زمان‌بری نمی‌باشد.

### 4.3.2 ایجاد تغییر در معماری شبکه

آنچه که در این پایان نامه به عنوان شبکه مورد استفاده قرار گرفت، شبکه‌های عصبی بازگشتی بود. اما انواع دیگر معماری‌های شبکه‌های عصبی نیز وجود دارد که می‌توان آنها را مورد بررسی نمود. شبکه‌های عصبی برگردنده<sup>۱</sup> از معماری‌هایی است که با استفاده از بازخوردهایی که از گره‌های میانی به گره‌های سطح پایین‌تر می‌گیرد کار می‌کند و می‌تواند جز معماری‌هایی باشد که کارکرد آن برای مجموعه داده معرفی شده در بخش 4.1 مورد بررسی قرار بگیرد. همچنین استفاده از معماری شبکه‌های عصبی کانولوشنی<sup>۲</sup> به جای شبکه‌های بازگشتی، معماری دیگری است که می‌تواند مورد استفاده قرار بگیرد و کارکرد آن مورد بررسی بیشتر قرار بگیرد.

### 4.3.3 استفاده از معماری موجود جهت پیش بینی سری‌های زمانی

پیش‌بینی<sup>۳</sup> سری‌های زمانی<sup>۴</sup>، فرآیند پیش‌بینی موقعیت‌های ناشناخته در آینده‌ی یک روند است. پیش‌بینی سری‌های زمانی می‌تواند با استفاده از تجربیات گذشته انجام شود و ابزار بسیار قدرتمندی است که توسط آن می‌توان برای موقعیت‌های غیر معمول، مانند افزایش غیر معمول تقاضا در مباحث اقتصادی، خود را آماده نمود [11]. استفاده از معماری RNTN جهت پیش‌بینی سری‌های زمانی بسیار ساده بوده و تولید مجموعه داده آن نیز چندان هزینه و زمان بر نمی‌باشد.

---

<sup>1</sup> Recurrent Neural Networks

<sup>2</sup> Convolutional Neural Networks

<sup>3</sup> Forecasting

<sup>4</sup> Time series

#### 4.4 نتیجه‌گیری

در این فصل، مدل‌های معرفی شده در فصل 3 مورد پیاده‌سازی و بررسی قرار گرفت. مجموعه داده‌ی مورد استفاده نیز در بخش 4.1 مورد معرفی قرار گرفت. در این بین، مدل مورد نظر ما (RNTN) بیشترین دقت را بین مدل‌های پیاده‌سازی شده به دست آورد و توانسب با افزایش 5.4٪ دقت، بیشترین دقت را در بین تمام مدل‌هایی که وجود داشته‌اند به دست آورد. همچنین عملکرد این مدل بر روی جملاتی که معکوس شده‌اند نیز مورد بررسی قرار گرفت که نشان داد از این نظر هم دقیق‌تر از دیگر مدل‌ها می‌باشد.

## References

- [1 G.-B. L. C. a. C. K. S. Huang, " Universal approximation using incremental constructive  
] feedforward networks with random hidden nodes," Vols. 879-892, 2006.
- [2 M. Y. G. B. a. D. J. E. Rafiq, "Neural network design for engineering applications," Vols.  
] 1541-1552, 2001.
- [3 O. a. C. C. Irsoy, "Deep recursive neural networks for compositionality in language," vol.  
] pp. 2096, 2014.
- [4 J. P. E. H. H. A. Y. N. a. C. D. M. Richard Socher, "Semisupervised recursive autoencoders  
] for predicting sentiment distributions," 2011.
- [5 C. C. L. A. N. a. C. M. Richard Socher, "Parsing natural scenes and natural language," 2011  
] .
- [6 A. P. J. Y. W. J. C. D. M. A. Y. N. a. C. P. Richard Socher, "Recursive Deep Models for  
] Semantic Compositionality Over a Sentiment Treebank," 2013.
- [7 D. K. a. C. D. Manning., "Accurate unlexicalized parsing," 2003.  
]
- [8 B. P. a. L. Lee, "Exploiting class relationships for sentiment categorization with respect to  
] rating scale," Vols. 2(1-2):1–135, 2005.
- [9 B. H. C. D. M. a. A. Y. N. R. Socher, "Semantic compositionality through recursive matrix  
] vector spaces," 2012.
- [1 A. P. J. W. J. C. C. M. R Socher, "Recursive Deep Models for Semantic Compositionality  
0] Over a Sentiment Treebank," 2013.
- [1 " Principles of forecasting," 2001.

1]

[1 o. Bengio, "Learning deep architectures for AI," 2009.

2]

[1 R. D. P. V. C. J. J. K. T. H. T. a. J. S.-t. oshua Bengio, " A neural probabilistic language  
3] model.," vol. In Advances in Neural Information Processing Systems, 2001.

[1 R. C. a. J. Weston, "A unified architecture for natural language processing: Deep neural  
4] networks with multitask learning," 2008.

[1 J. W. L. B. M. K. K. K. a. P. K. Ronan Collobert, "Natural language processing (almost)  
5] from scratch," vol. 12:2493–2537, 2011.

[1 J. L. Elman., "inding structure in time.," 1990.

6]

[1 E. H. H. J. P. C. D. M. a. A. N. Richard Socher, "Dynamic pooling and unfolding recursive  
7] autoencoders for paraphrase detection," 2011.

[1 J. S. L. complex, "extended sequences using the principle of history compression," 1992.

8]

[1 S. E. H. a. Y. Bengio., "Hierarchical recurrent neural networks for long-term dependencies,"  
9] 1995.

[2 M. H. a. B. Schrauwen, "Training and analysing deep recurrent neural networks".

0]

[2 P. S. a. P. F. Yoshua Bengio, "earning long-term dependencies with gradient descent is  
1] difficult," 1994.

[2 C. G. a. A. Kuchler, "earning task-dependent distributed representations by backpropagation  
2] through structure," vol. 1, 1996.

[2 A. B. a. Y. B. Xavier Glorot, "Deep sparse rectifier networks," 2011.

3]

[2 Q. V. L. a. T. Mikolov., "Distributed representations of sentences and documents," 2014.

4]

[2 E. G. a. P. B. Nal Kalchbrenner, "A convolutional neural network for modelling sentences," 5] 2014.

[2 I. S. K. C. G. S. C. a. J. D. Tomas Mikolov, "Distributed representations of words and 6] phrases and their compositionality," 2013.

[2 N. S. A. K. I. S. a. R. R. S. Geoffrey E Hinton, "Improving neural networks by preventing 7] co-adaptation of feature detectors.," 2012.

[2 I. S. a. G. E. H. Alex Krizhevsky, "magenet classification with deep convolutional neural 8] networks," 2012.

[2 T. N. S. a. G. E. H. George E Dahl, "Improving deep neural networks for lvsr using 9] rectified linear units and dropout," 2013.

[3 E. H. a. Y. S. John Duchi, "daptive subgradient methods for online learning and stochastic 0] optimization," 2011.

[3 Statsoft, "Time Series Prediction in ST Neural Network," 2002.

1]

[3 "پیش‌بینی (اقتصاد)" [Online]. Available:

2] [https://fa.wikipedia.org/wiki/%D9%BE%DB%8C%D8%B4%E2%80%8C%D8%A8%DB%8C%D9%86%DB%8C\\_\(%D8%A7%D9%82%D8%AA%D8%B5%D8%A7%D8%AF\)](https://fa.wikipedia.org/wiki/%D9%BE%DB%8C%D8%B4%E2%80%8C%D8%A8%DB%8C%D9%86%DB%8C_(%D8%A7%D9%82%D8%AA%D8%B5%D8%A7%D8%AF)).

[3 D. K. a. C. D. Manning., "Accurate unlexicalized parsing," 2013.

3]

