A TENSOR-COMPETITION BASED ARCHITECTURE: TO CAPTURE THE INFLUENCE OF WORD SENSE

Eshaa M. Alkhalifa

Dept. of Comp. Sc., College of IT, University of Bahrain, Isa Town, Bahrain ealkhalifa@itc.uob.bh

ABSTRACT

Studies done in the field of word meaning, and in particular the Latent Semantic Analysis (LSA) project, found out the representation of words as vectors in a high dimensional semantic space. However, the system completely neglects the effect of word order on this representation. This results in a semantically viable model that neglects the differences in sentence structure. If the sense a word implies in a sentence is different, as with "the letter is sealed" versus "the letter is N" then LSA assumes them to still be similar. The following model takes LSA data and processes them in tensor space to further incorporate the aspect of word sense. The architecture is amazingly robust and is sensitive to how a word is used in different settings. Ten rules are tested to show how it can add to the representation of meaning offered by LSA.

1. INTRODUCTION

Latent Semantic Analysis (LSA) is a powerful high dimensional semantic model that ignores the effects of directionality on semantics. It treats contexts of around 2000 words as bags of words ignoring any word order by studying word use in context. The assumption is that a word is in some way semantically described by the words that appear alongside it in the same context.

Landauer and Dumais [6] created a multidimensional vector representation in "semantic spaces". The process starts out with a co-occurrence matrix that has as entries the number of times each word appears in a particular context. Then these numbers are modified with a special function that takes the "entropy" or importance of the word into consideration and takes its log. Then the adjusted matrix is taken through a Singular Value Decomposition to transform it into three matrices. The first gives a vector representation of the words in terms of the contexts. The second gives a vector representation of the contexts in terms of the words. The third is a scaling matrix that only has nonzero elements along the diagonal. This scaling matrix has constant values to ensure that when the three are multiplied then the original matrix is obtained. For example if the three resulting matrices are W, P, S with S for the scaling matrix then the original would be obtained when we multiply W*S* P'.

> ***"We postulate that the power of the model comes from dimensionality reduction"[6]***

The secret of LSA lies in that before the vector components are put together again, the number of columns with values is reduced to usually 300 dimensions. When this is done, the matrix that results from the multiplication process, redistributes the effects of these numbers over a large span of cells. Altering a single value in the original matrix, results in a change of a large group of cells in the resulting matrix. Results show a high degree of correlation between the predictions made by the model and human behavior. It also seems to have the ability of extracting a vector representation that is capable of assessing the "semantic" distances between words in a contextual space. The semantic space LSA uses to represent vectors, is a "world of words" where each word has a location based on the distance its meaning is from other words. This distance is estimated without any regard to where the word appeared in the sentence, nor does it accommodate for multiple possible "senses".

So, it should not be surprising that it confuses different "uses" or "instances" of words such as "the letter is sealed" and "the letter is N". It should be clearly evident that the instance that is invoked into a reader's memory for the word "letter" is different. In the first, it would imply something written on a paper and placed inside an envelope and in the second it would imply a letter of the alphabet. Actually, the number of words that a language uses is not unlimited, and the importance of the contextual space comes from the order of these words. Needless to say that the role attributed to a word in a sentence is to be a part of a whole, related not only to the words behind or ahead, but related to the sentence itself as part of the meaning.

In order to further understand the problem we should look a little deeper into the makings of LSA. It attempts to capture meaning based on the structural making or word co-occurrence within contexts. The location of the

word in the sentence or the syntactical category it belongs to, has no bearing on the process of extracting meaning. This led researchers as Wiemer-Hastings [9] to further investigate the effects of the neglected syntax and attempt to incorporate it into the LSA framework. He separated the sentences into atomic clauses or propositions and then segmented them by hand to break them into strings composed of subject noun phrase, verb and object noun phrase. Antecedents were used to resolve pronouns and conjunctions were dealt with by distributing the arguments. Then he attempted to evaluate the similarity of this presentation using a variety of measures. Results showed that the best approach to combine the similarities of the sentence parts is non-linear and even that wasn't as close to human judgment as LSA. Wiemer-Hastings and Zipitria [8] then went on to a further test, incorporating syntax through two methods. The first was to tag the words used for the training corpus at 100,200, 300 and 400 dimensions and this did not produce any favorable results. Then they tried a structured LSA or SLSA where they broke up sentences into parts as was shown above and used that as training material to find results that correlate slightly better than LSA which does not pay any attention to sentence structure.

However, explicit, breaking up of a sentence into its component structure is not the only approach followed towards incorporating syntax into word meaning. The Hyperspace Analogue to Language (HAL) is similar to LSA in that it is based on co-occurrences of words, except that word order information is retained [4]. For each window of 5 or 10 words a table is formed to indicate the words that precede it and those that follow it. This information is then concatenated into a vector that can be up to 100 dimensions long. The main difference between LSA and HAL is that the former uses around 2000 word passages to infer co-occurrence and uses singular vector decomposition to re-distribute the data utilizing only the highest 300 influential factors. The latter, uses the co-occurrence data as is, and uses a window onto the text and then forms the vectors through concatenation. Both methods seem successful and compete quite well with each other.

So what is there to lose if some word order sensitivity is added to LSA, through a post-processing step of the data?

The context of the same word can change, and when it changes, a different sense of the word can be implied. It is these cases that LSA seems to neglect along with word order. In order to incorporate word order into a vector based model, the model must be raised one rank to become a tensor-based representation.

2. SOME TENSOR MATH

The moment a person starts to think about a world that cannot be imagined or drawn onto a piece of paper, anxiety starts to knock the door. Yet once the tensor world is described as follows, everything starts to make sense.

In a scalar field, a single number describes a point, while in an n-dimensional vector field, n-numbers are needed to describe a point. In a tensor field n-squared numbers are used to describe a point or n-cubed numbers, etc.

In other words, our tensor representations would be in the form of matrices because they add only one rank to vectors. In the world proposed here, nothing is done or changed in the original LSA world of vectors, other than simply representing words as n-dimensional vectors in that world. The difference only shows when one is to study or analyze sentences, because these, now have tensor representations that are matrices composed of a number of rows each for the words in that sentence in the order they appear left-to-right.

THE	0.99	0.78	1.00	0.80	0.96	0.85
FLAG	0.29	0.17	0.31	0.27	0.32	0.19
FLIES	0.26	0.23	0.24	0.24	0.24	0.23
HIGH	0.45	0.33	0.46	0.33	0.45	0.40

Table 1: Four words are shown, each described in 6-dimensional space

THE	FLAG	FLIES	HIGH		
0.99	0.78	1.00	0.80	0.96	0.85
0.29	0.17	0.31	0.27	0.32	0.19
0.26	0.23	0.24	0.24	0.24	0.23
0.45	0.33	0.46	0.33	0.45	0.40

Table 2: The matrix as a whole describes the sentence maintaining the order of the words.

3. A TENSOR BASED-COMPETITIVE ARCHITECTURE

This idea stems from the architecture of a self-organizing neural network algorithm, designed by Tuevo Kohonen that allows the inputs to organize themselves in a high dimensional space as per a competition [5]. Input patterns are presented through a training phase and each is classified by the units it activates based on its similarities to other members of that group. These similarities are mapped into 'closeness' measure that is dynamically estimated by the competitive layer. The networks are unsupervised in the sense that they are only given the inputs and allowed to search for 'closeness' measures between the different inputs. The decision of how close any input is, must be assessed through a special function that can take various different forms or ways to measure the differences between the points. The most common of which is to just get an estimate of how far the

vectors are from each other through a function that is similar to this:

$$\underline{V} - \underline{W} = \sum_{i=1}^{2616} V_i - W_i$$

Note that the subscripts are used to show that here the vectors are in 2616-dimensions of space.

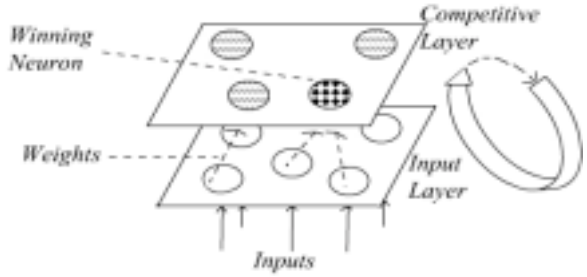

Figure 1: A Competition Based Network Architecture

Since this is the basic architecture used for competitive networks, it shows itself as an ideal candidate for applying tensor mathematics without adding too many complications.

3.1. Basic Materials

In order to do that, the vector operations required by the process must be replaced with tensor operations. Weights have to be three-dimensional as well in order to enable them to partition the tensor space. The words used in this model were collected from rules used in 68 runs in the Wason selection task and then the values of their representations in terms of each other were found through LSA's online site. This resulted in a matrix of 218x218 words, such that each word has a descriptive vector or length 218. Then ten rules were selected from the group such that their lengths do not exceed 12 words in order to ensure comparable sentence length. These rules are as follows:

R30=If I go to Leeds, I travel by car.
R31=if I go to Leeds, I travel by train.
R32=if I go to Manchester I travel by car.
R33=if I go to Manchester I travel by train.
R38=If the tablecloth is brown then the wall is white.
R42=the tablecloth is brown only if the wall is white.
R41=If a letter is sealed, then it must carry a 20-cent stamp.
R37=if the letter is an N, then the number is a 3.
R26=if an envelope is sealed, then it must have a 1-mark stamp
R14=if a house was built before 1979 then it has a fireplace.

The rule numbers are shown because the target of the study is to eventually include all 42 rules into one model and add to that some extra features to produce a predictive model for that task. However, ten rules are sufficient for the purposes of testing the viability of the architecture because it will result in around 497040 repetitions for the task described here with a limit set to 19 epochs. It should be clear at this stage that this architecture is not one that requires few computer resources. However, it will also be shown that its results are equally powerful.

3.2. More Tensor Math

The inputs are presented to the network and it has to make certain decisions based on how "close" the weights are to the inputs. Distances between tensors can be estimated using the following equation:

$$\underline{\underline{T}} - \underline{\underline{U}} = \sum_{i=1}^{12}\sum_{j=1}^{218} T_{ij} - U_{ji}$$

The result here is the distance between the two in tensor form so it is a little difficult to compare the results to each other and select the closest weight vector. Another operation is required to estimate the magnitude of the tensors and selecting between them. The following equation is used to estimate the magnitude:

$$\left|\underline{\underline{T}}\right| = \sqrt{\frac{1}{2}\sum_{i=1}^{12}\sum_{j=1}^{218} T_{ij}^2}$$

From there a winning tensor is selected which is usually the one with the lowest magnitude and the associated weight tensor is adjusted. Another equation that was necessary to include is that of the dot product. The reason is that the dot product in tensor math is not similar to that in matrix operations as surely is evident in the other operations shown here. The formula used in the testing phase is therefore as follows:

$$\underline{\underline{T}} : \underline{\underline{U}} = \sum_{i=1}^{12}\sum_{j=1}^{218} T_{ij} U_{ji}$$

3.3. Implementation Issues:

The architecture was implemented and tested using MATLAB. The only main issue here is processing time requirement that seems to grow exponentially with the size of the tensor representation, but not with the number of sentences shown to the system, which is not bad.

4. TEST RESULTS

Since the model presented here aims at showing the effect of this type of architecture and its ability to preserve the effect of order on meaning, results had to be compared to those obtained from LSA. The sentences or rules shown above were also run through LSA's online site to obtain a similarity matrix. This matrix was slightly adjusted to be shown here by subtracting all values from 1 in order to have a 0 value for the same items as with the output of the system. The result is shown here:

	R30	R31	R32	R33	R38	R42	R41	R37	R26	R14
R30	0									
R31	0.07	0								
R32	0	0.07	0							
R33	0.07	0	0.07	0						
R38	0.44	0.41	0.44	0.41	0					
R42	0.46	0.44	0.46	0.44	0.01	0				
R41	0.49	0.46	0.49	0.46	0.42	0.42	0			
R37	0.56	0.54	0.56	0.54	0.47	0.46	0.14	0		
R26	0.47	0.45	0.47	0.45	0.38	0.38	0.19	0.18	0	
R14	0.38	0.36	0.38	0.35	0.3	0.31	0.38	0.43	0.35	0

The model on the other hand resulted in a representation for each of the sentences that was then post-processed to result in the following table:

	R30	R31	R32	R33	R38	R42	R41	R37	R26	R14
R30	0									
R31	8.8	0								
R32	8.5	1.7	0							
R33	17.2	8.5	8.8	0						
R38	144.1	135.3	135.8	127	0					
R42	147	138.3	138.6	129.9	19.5	0				
R41	243.8	235.1	235.5	226.7	100.1	98.7	0			
R37	311.5	302.7	303.1	294.3	167.9	165.1	68.6	0		
R26	266.8	258	258.5	249.7	123	121.4	23.1	45.8	0	
R14	288.3	279.5	280	271.2	144.7	142.5	45.2	24.6	22.5	0

The means in the two matrices are 0.3202 and 135.3. The important difference though appears through the extra distancing or bringing closer that the NN model added to the LSA data that was used. Groups were formed, finding a high degree of similarity between the rules R30, R31, R32 and R33. The model placed these four at a distance from the rest identifying their high level of similarity but did not regard any of them as identical to the other. The model also grouped, although not so closely the rules R41, R37, R26, R14 into a group of their own. The third group was the one that contained the rules R38 and R42, which have the opposite structure shown as very close with the shape of the letter "X" when expressed through a graph, crossing at the center around where the words "only if" would appear.

The LSA data, on the other hand does not seek similarities in order to group the rules but it did show the rules R30, R31, R32 and R33 close to each other. R38 and R42 were superimposed on each other and the R26, R37 and R41. The Rule R14 was alone.

As for the correlation values between the rules, we notice that LSA found rules R30 and R32 as identical, and rules R31 and R33 as identical. This implies that the words "Leeds" and "Manchester" mean exactly the same thing in both. The tensor model seemed more sensitive to this and kept them apart although relatively close to each other. To further emphasize this, the tensor model found a bigger distance between rules R30 and R33 because they differ in two words "Manchester" vs "Leeds" and "car" vs "train".

When we compare the rules R38 and R41 which are identical except for the form, we find that LSA gives a 0.01 difference while the tensor model finds a bigger difference between them. This added sensitivity was the main reason behind suggesting this post-processing step.

5. WASON SELECTION TASK STUDY

This work has come as part of a study of word meaning aimed at identifying influential factors in the Wason selection task [7]. The original task involved showing subjects four cards with a letter on one side and a number on the other. The cards classically contained, A,4,B,7 and the subjects were told to observe the rule: if there is an A on one side, then a 7 must be on the other side. They would then indicate which cards they wish to turn over to check if this rule is not broken. A great deal of work went into the investigation on the low achievement levels of students in this task and various forms of rules were utilized. A previous study of a pool of reported experiments along with gathered data about the rules themselves, lead to identifying two implied factors by the rules in order to be strongly correlated with subject performance [1]. These two factors were "temporal distancing" and egoism which is implied when the subject is addressed as "you are..". Another study using conveyor systems showed that the implication of a moving system or a static system that is capable of moving strongly affects subject behavior [2]. Yet another study by placing the cards and conveyor in colors, shows a Stroop like interference, which hints that this "directional" effect may be semantic [3]. Therefore, the most challenging data to test this system, would be to use the same rules that challenge researchers who works in the Wason task worldwide. The rules used here were

some of those in the first study [1] and the reported correct student choices were as follows:

R30=62.5% ***R32=62.5%***
R31=62.5% ***R33=62.5%***

Note that the above four were grouped together by the tensor model and placed close to each other by LSA so they both did equally well in recognizing the similarities between them.

R38=15% ***R42=4%***

LSA did not recognize any differences between the above rules while the tensor model did recognize a moderate difference. The tensor model then was capable of predicting a difference in subject performance.

R41=86.5% ***R37=6.25%***
R26=80% ***R14=40%***

LSA isolated the R14 rule from the group while the tensor model grouped the four together. Ordinarily if one would be isolated it should be R37, so neither the performance of LSA nor the model was appropriate with this group of rules. Although the tensor model does improve on LSA's performance it is not to a satisfactory degree.

6. DISCUSSION

The results show that although the LSA model is extremely powerful, it has neglected word order and its influence on meaning. The tensor model proposed, is capable of adding this feature to a LSA based matrix simply by maintaining word order within the comparable set of rules. The first word is compared to the first word and the second to the second. It may even seem elementary except that in order to do that, one has to use tensor mathematics. This math has enabled this model to be created and proved to be simpler than one may think.

Look at rules R41 and R37, the first has "If a letter is sealed" and the second "If a letter is an N". Both have the same word "letter" but each instance of it carries a different sense of the word. LSA found these two highly similar with 0.14 measure and they are closer to each other than R41 is to R26, which mentions the word "envelope". On the other hand, the tensor system, places R41 (which has the sealed letter) to R26 (which has the sealed envelope) than it is to R37 (which has the letter N). It seems that this model adds more than just word order as it seems to be capable of aligning verbs with each other within the construct of similar sentences to be able to extract word "sense" information. Further testing is of course needed.

A second positive result is that this model can be generalized to all applications where structures are to be compared and classified through a self-organizing model. The formulas do not need to be changed except to accommodate for the sizes of the matrices and essentially the same NN program can be used. It can for example, start to propose some of the groupings that seem evident in Wason selection task [7] performance measures obtained by various researchers around the world but not adequately. There is still much more to do.

7. REFERENCES

[1] E.M. Alkhalifa, Egoistic reasoning 'In Time' in the Selection Task, Proceedings of the Third International Conference on Cognitive Science, Beijing, China, pp.109-113, 2001a.

[2] E.M. Alkhalifa, Directional thought effect in the Selection Task, Proceedings of the Third International Conference on Cognitive Science, Beijing, China, pp. 171-176, 2001b.

[3] E.M. Alkhalifa, The effects of meaning in a white/black setting on reasoning, The 6th Australasian Cognitive Science Society, OzCogSci'02, Fremantle, Western Australia, 2002.

[4] C. Burgess, K. Lund, Modeling parsing constraints with high-dimensional semantic space, *Language and Cognitive processes*, v. 12, pp. 1-34, 1997.

[5] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, 1995.

[6] T. Landauer, S. Dumais, A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, v.104 pp. 211-240, 1997.

[7] P.C. Wason, Reasoning, In B.M. Foss (eds), New horizons in Psychology, Harmondsworth, UK: Penguin, 1966.

[8] P. Wiemer-Hastings, I. Zipitria, Rules for syntax, vectors for semantics, Proceedings of the Twenty Third Annual Conference of the Cognitive Science Society, Mahwah, NJ: Lawrence Erlbaum Associates, 2001.

[9] P. Wiemer-Hastings. Adding syntactic information to LSA, Proceedings of the Twenty Second Annual Conference of the Cognitive Science Society, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 989-993, 2000.