Automated Scoring of Levels of Integrative Complexity Using Machine Learning and Natural Language Processing

A. Kannan Ambili and K. Rasheed
Institute for Artificial Intelligence, University of Georgia, Athens, GA, USA

Abstract- *Conceptual/Integrative complexity (IC) is a construct used in political psychology and clinical psychology to gauge an individual's ability to consider different perspectives on a particular issue and subsequently form a conclusion that draws from the said perspectives. Presently IC is scored from text manually which is time-intensive, laborious and expensive. For a rater to be qualified to score IC, it is standard that he/she go through a rigorous training program. Consequently, there is a demand for automating the scoring, which could significantly reduce the time, expense and cognitive resources. Any algorithm that could achieve the above with a reasonable accuracy could assist in researchers who are interested in broadening the horizon for IC research. Furthermore, such a development could also assist in the design of intervention systems for reducing the potential for aggression, systems for recruitment processes and even training personnel for improving group complexity in the corporate world. In this study we used machine learning and natural language techniques to predict IC levels from text. We developed an intelligent feature called Semantic Paragraph Coherence for the prediction of IC levels in text. We achieved over 83% accuracy in a three way classification.*

Keywords: Integrative complexity, multi-class classification, semantic similarity, natural language processing

1 Introduction

Conceptual/Integrative Complexity is a construct in psychology that measures in a particular sample of text or speech the extent of differentiation and integration exhibited by the author [1]. Differentiation is the author's ability to examine differing perspectives on an issue; the higher the number of perspectives being examined on a particular, the higher is the differentiation. Integration refers to the author's skill in considering the possibly connected perspectives at hand and using these connections to form well-reasoned conclusions.[1] It has been claimed as the most used and widely validated measurement of complex thinking. Regardless of the content contained in a text sample, IC is a measure used to capture the cognitive strategies used to formulate the structure of thought of the author.

IC has been used to predict aggression in political psychology [2]. It has also been found to be an efficient predictor of performance and corporate social responsibility [3.4]. Studies have found that liberal or left-leaning politicians often tend to have high IC [2] [5] [6] Decision making can sometimes be hampered due to high IC [7]. Considering all these applications, it seems to be of immense importance that an efficient automated scoring method for IC be developed.

1.1 The need for Intelligent Features

The fact that the level of IC in a text is contingent upon the relationships that connect different perspectives could be used as a heuristic in determining the level of IC. In machine learning, the problem could be better solved with the consideration of predictors (features) that indicate in text, the amount of differentiation and integration. However, current literature has not defined algorithms that can accurately measure said constructs in an efficient manner. Linguistic Inquiry and Word Count (LIWC) [8] is a program that counts words that belong to two psychologically meaningful categories: exclusion words and conjunctions. Exclusion words (e.g. but, without, exclude) are helpful in making distinctions among different sentences. Conjunctions (e.g. and, also, although) join multiple sentences and contribute to measuring Differentiation [9]. The authors of the current body of work had performed research on the automation of the scoring of IC and were successful in obtaining accuracies of approx. 78% [10]. The authors hypothesized that the inculcation of a pre-designed NLP feature in the previously adopted machine learning methodology could improve performance and accuracy. This body of work focuses on proving that particular hypotheses as well as on improving the performance of the automated scorer for Integrative Complexity. While most text classification

problems are easily solved through a bag of words approach, this particular problem requires a deeper understanding of the interlinking of arguments (or in other words, perspectives) in a given fragment of text. Consider the example given below (Taken from Peter Suedfeld's integrative complexity training workshop [1]):

> *"Advances made in the chemistry of antiseptics and the techniques of surgery are not wholly responsible for the new standards of lifesaving in war. An alert and courageous system of fully equipped yet highly mobile surgical units following close behind the assault troops has resulted in an immense saving of time between the battlefield and the operation table. In surgery time-saving is akin to lifesaving."*

The thesis for the instance is that '*the new standards of lifesaving in war*' cannot be just attributed to '*antiseptics and the techniques of surgery*'. Following it, is a contributing perspective that "*the alert and courageous system of fully equipped yet highly mobile surgical units following close behind the assault troops*. Subsequently, the author makes the differentiation more substantial with the declaration that this has "*resulted in immense saving of time*". This differentiation is immediately followed by the integration-bearing declaration that "*time-saving is akin to lifesaving.*" thereby giving the thesis further support. Since there is minimal differentiation and integration, this text sample could be scored as having moderate Integrative complexity, which is equivalent to a score of 3- 5.

Therefore for a text to be qualified as having high integrative complexity, numerous differentiations have to be made, subsequently followed by integration. In other words, differentiating statements relate to each other with a non-zero amount of semantic similarity. Most differentiating and integrating statements would intuitively be semantically similar in content to an extent. In this particular example, the differentiating statements do have some semantic similarity. The semantic content of the reference made in the thesis sentence "*lifesaving in war*", is referred to semantically in content in the subsequent differentiating statement as "*the assault troops*" and "*the battlefield*" and "*immense saving*". In the final integrating conclusion, we can determine a semantic similarity to "*lifesaving*". It is this property that is exhibited by the 'integrative-ly complex' that could be exploited in the prediction of levels of integrative complexity.

1.2 Semantic Paragraph Coherence as a feature

Measuring semantic similarity between sentences could be translated into measuring the semantic similarity of words that carry the most information in these sentences. Most often the semantic content in sentences comes from the nouns, verbs and adjectives and to a lesser degree on adverbs, prepositions and the rest. Traditionally, semantic similarity between sentences would be limited to analyzing the similarity between shared words [11], which worked reasonably well in texts of longer lengths. However, for shorter texts, a method which focused on the semantic meaning of the word rather than the word itself was required.

1.2.1 Semantic similarity between words.

The method for calculating semantic similarity between words in this paper is based on Li, Bandar & McLean's work in 2003 [12], where the similarity of two words is calculated using a hierarchical semantic knowledge bases (e.g. WordNet [13][14][15] The work presented in this paper calculated semantic similarity as a function of path length (the minimum number of words lying between the considered words in the hierarchical knowledge base) and depth (the depth of the subsumer in the hierarchy). Path length and depth are both derived from a lexical knowledge base. α and β are parameters that are used to scale the contributions of path length and depth respectively. Let the semantic similarity between two words w_1 and w_2 be noted by $S(w_1, w_2)$. Then according to the word similarity measure proposed in [12]:

$$S(w_1, w_2) = f(l).f(h) \qquad (1)$$

In the above measure, $\alpha \geq 0$ and $\beta > 0$. The proposed optimal values are: $\alpha = 0.2$, and $\beta = 0.6$[12].

1.3 Implementation Details

This section describes the method used to calculate the semantic similarity between two words. Only a brief account is given here, for further explanation, please refer to the original paper [12] [16]. The method was coded in SWI Prolog [17], since WordNet version 3.0 [13] [14] [15] was also available in Prolog.

1.3.1 Contribution of path length

The path length between two words in a hierarchical knowledge base can vary between 0 to large numbers. Hence the function should be designed so that it will have values ranging from 0 to 1. This function will depend on three cases: In the first case, $f(l) = 1$; if w_1 and w_2 belong to the same concept. In the case that the two words do not belong to the same concept, but have the same word linking them, their semantic similarity is calculated as:

$$f(l) = e^{-\alpha l} \qquad (2)$$

l is the sum of the number of words leading up from both words to the same word.

1.3.2 Contribution of depth

The first common hypernym between w_1 and w_2 is called the subsumer of w_1 and w_2. h is the depth of the subsumer in the hierarchical semantic nets. For example consider the path between 'boy' and 'girl', the path is 'boy-male-person-female-girl', then 'person' is the subsumer for 'boy' and 'girl'. The depth

is calculated by counting the levels from the subsumer level to the top of the lexical hierarchy. The subsumer of the shortest path is considered in deriving the depth of the subsumer in case of polysemous words.

$$f(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \qquad (3)$$

1.3.3 Calculation of semantic similarity between words

The semantic similarity between two words w_1 and w_2 be noted by $S(w_1, w_2)$ (i.e a product of (1) and (2)) [12]:

$$S(w_1, w_2) = f(l).f(h) = e^{-\alpha l}.\frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \qquad (4)$$

1.4. Calculation of Semantic Paragraph Coherence

Our work proposes the use of Semantic Paragraph Coherence as a feature to predict Integrative Complexity. Scoring of Integrative complexity involves scoring the text by determining the levels of differentiation and integration. A text that has been scored extremely low on Integrative Complexity can be seen as a series of unconnected discourse, or as a paragraph that focuses on a single thesis with descriptive statements. Such a fragment of text wouldn't necessarily make references to the semantic information present in the thesis later on in the paragraph to discuss different perspectives or come to a well-reasoned plausible conclusion. It is this assumption that is behind the development of the proposed feature.

The proposed method calculates Semantic Paragraph Coherence by calculating the semantic similarity between the first sentence in a sample text and the rest of the sentences in the sample text. The calculation of semantic similarity between the sentences is limited to nouns and verbs. The assumption behind this choice is that nouns and verbs carry the most semantic information, and at the same time this keeps the number of calculations to a smaller number, thereby reducing computational complexity.

The calculation of Semantic Paragraph Coherence is a two-step process. Initially, the calculation of all the semantic similarities of the words in the first sentence with every other word in the rest of the sentences in the paragraph is performed. This step itself is composed of two steps. For each word, w_j present in the first sentence (otherwise named as the topic sentence), the semantic similarity between itself and every relevant word, w_i in the rest of the paragraph is calculated. Let this value be s_{ij}. Here m is the maximum value of i, *i.e.* the total number of relevant words present in the paragraph (with the exception of the topic sentence). Therefore for a word w_j present in the topic sentence, the associated semantic similarities with the rest of the paragraph is formulated as below. Let $g(w_j)$ be this measure. Then:

$$g(w_j) = \frac{1}{m}\sum_{i}^{m} s_{ij} \qquad (5)$$

Let n be the total number of relevant words in the topic sentence. Then the total associated semantic similarity value of the paragraph could be treated as, sum.

$$sum = \sum_{j}^{n} g(w_j) \qquad (6)$$

Semantic Paragraph Coherence, P could be calculated as:

$$P = e^{-1*sum} \qquad (7)$$

Consider the below statement, taken from Dr. Suedfeld's Integrative complexity training workshop page [1]:

> *' The experience of life's hardships and comforts fosters an awareness of both the value and impermanence of the moment ; all of these influence and are influenced by the meaning we make-which is further negotiated over time and through interaction with others-and manifested in our autobiographies..'*

The sample text (shown above) scores high on Integrative Complexity. Our proposed method scored a P value of 0.30. Whereas the text given below [1] scores low on Integrative complexity. And has a P score of 0.930:

> *'So much for my apologies. There are plenty of them, perhaps too many. Were it not for your letter I should feel myself almost guiltless. But since you apparently went on thinking about the purse and possibly even searching for it, all apologies are of course inadequate and I must resort to asking you not to spoil my pleasure in finding the purse, by being angry with me for my negligence. For that would be-even though the purse contained 900 crowns (which may explain my haste in telling you) a tremendously high finder's fee which I would be obliged to pay to lucky chance. You won't do that I'm sure."*

It has to be noted that for some samples Semantic Paragraph Coherence may not be the best predictor. These instances could be identified as outliers. From these examples, it could be inferred that Semantic Paragraph Coherence could act as a predictor for scoring Integrative complexity with non-zero error.

2 Approach

2.1 Data Selection and Experimental Setup

The data for the project consisted of 83 text samples along with the scores provided by manual scoring by trained coders. The data was taken from Suedfeld's Complexity Materials Download Page [1], where the data has been made available free for download for scorers who want to practice scoring.

Each instance has been scored on a 1-7 scale. The first step in Pre-processing involved binning the instances into three bins. Instances that have been given IC scores of 1 or 2, were classified as having low IC and therefore given a class label of 'low'. Similarly instances that have been scored IC scores of 3, 4 or 5 were classified as having medium levels of IC, and were given a class label of 'mid'. Subsequently, instances that were scored IC scores of 6, 7 were classified as having high IC and were given class labels 'high'.

The code for extracting the value of the Semantic Paragraph Coherence feature was written in SWI Prolog [18]. The code made use of WordNet 3.0 [13] [14] [15] written in Prolog to design the feature. Then the code was run on each instance to calculate the Semantic Paragraph Coherence of each instance. The code for calculating the length of a paragraph (in words) was also calculated in Prolog.

Then, the data was cleaned and converted into an ARFF (Attribute Relation File Format) file format for use in Weka. [17]. Feature selection methods played a huge role in this text-classification problem. Using the String to Word Vector filter in Weka [17], the string in the text attribute of each instance is converted to a set of attributes representing word occurrences, where each word is converted to lowercase before processing. Along with the bag of word features, we also included the Semantic Paragraph Coherence measure and length of the text sample. The number of attributes were reduced significantly using Attribute Selection methods.

2.2 Learning Methods

The project used several machine learning algorithms for experimenting with the data. For this purpose, the open source machine learning software, Weka [17] was used. The algorithms that are mentioned here, are the ones which have reported some of the best performances. They are Bagging, the Multinomial Logistic Regression model, Multi-layer perceptron, AdaBoost.M1 and the Multi-class classifier.

Adaboost (short for Adaptive Boosting) is a boosting algorithm that can be used to significantly improve classifier performance given that its weak learners can predict with a rates a little better than random guessing. A weak learning algorithm is run on different parts of the distribution of the training data and then combined to form a composite classifier, this is the basis of boosting [19]. AdaBoost.M1 is a special case of AdaBoost where easy examples that are correctly classified by the weak learning algorithms are given less weightage than examples that get misclassified by the weak learning hypotheses.

The Multinomial Logistic Regression Model is often used in Natural Language Processing applications because they do not assume statistical independence of features, as is often the case with text. The model is a generalization of the logistic regression model for multi-class problems. The probabilities describing the outcomes of an instances are modeled as a function of its features, using a logistic function.

Another classifier that we experimented with was the Multi-class classifier- suitable for the multi-class classification problem. The Meta classifier used binary classifiers to solve the 3 –class classification problem. The binary classifiers used for experimentation were the logistic regression and the multi-layer perceptron. Popular multi-classification methods like 1-against-1 and pairwise classification were used.

The Multilayer Perceptron (MLP) was also used in the experimentation part. An MLP consisting of multiple layers of nodes in a directed graph, uses a supervised learning techniques called backpropagation for training the classifier. The MLP used in this work contained only nodes that had sigmoid functions as activation functions. The learning rate was set at 0.3 and momentum was set at 0.2.

Bagging (also known as Bootstrap Aggregation) is ensemble meta-learning algorithm that is used to reduce variance and over-fitting. This algorithms grants 'votes' to base classifiers that are trained on different bootstrap samples. A final classifier is built from all the base classifiers trained on all the bootstrap samples, whose prediction is based on the most predicted by its base classifiers.

3 Evaluation

The performance of the multi-class classification methods is tested through stratified 10-fold cross-validation. Considering the limited amount of data, especially in the context of a multi-class classification problem, the standard way of predicting the error rate of a learning technique is to use stratified 10-fold cross-validation. Classification accuracy has been used as one of the performance measures for this problem. However emphasis should be given to performance measures such as precision, recall and F-1 measures, as they tend to be better measures when evaluating small classes (Manning et al., 2008).

4 Results

Results obtained were promising. Table-II shows the classification accuracies and r. Table-I show the precision, recall and F-1 measures of the classifications. Overall, higher values for classification accuracies and effectiveness measures have been reported for the proposed approach. The highest classification accuracy was reported by the Multinomial Logistic Regression Model with a ridge estimator-II. The same classifier also reported the highest precision and recall. Some of the classifiers have high precision (1.000) for the class high. While some of them have high recall for the class mid. Classification accuracies of 80% to 83% are at par with the human rater reliability of 80%. But since the dataset is relatively small, more focus should be given to the Precision, Recall and F-1 measures.

The addition of the Semantic Paragraph Coherence and length of the text sample as features have influenced the performance of the algorithms in a positive manner. The combination of these features along with a bag of word approach have produced a decent performance. Experiments conducted to evaluate the contribution of the newly designed feature produced favorable results. Results showed that the Semantic Paragraph Coherence feature was able to assist in the three-way classification.

Table I: Precision, Recall and F-1 measures for Bagging, Multi-Class Classifier and Multinomial Logistic Regression Model with a ridge estimator-II

Class	Bagging			Multi-Class Classifier			Multinomial logistic regression with a ridge estimator-II		
	Precision	**Recall**	**F-1 measure**	**Precision**	**Recall**	**F-1 measure**	**Precision**	**Recall**	**F-1 measure**
low	0.800	0.500	0.615	0.923	0.500	0.649	0.857	0.500	0.632
mid	0.790	**1.000**	0.883	0.762	0.980	0.857	0.803	**1.000**	0.891
high	**1.000**	0.600	0.750	0.857	0.600	0.706	**1.000**	0.800	0.889
Weighted. Avg.	**0.818**	**0.807**	0.790	**0.820**	0.795	0.779	**0.843**	**0.831**	**0.816**

Figure 1: Performance comparison for multinomial logistic regression model-II (with a ridge estimator)

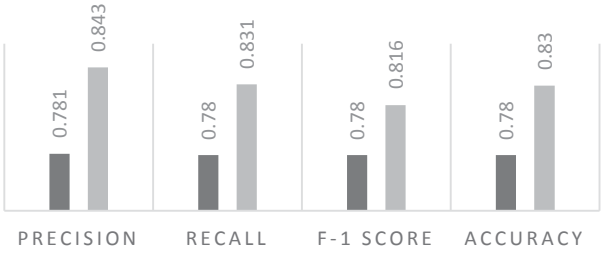

Figure 2: Performance comparison for bagging

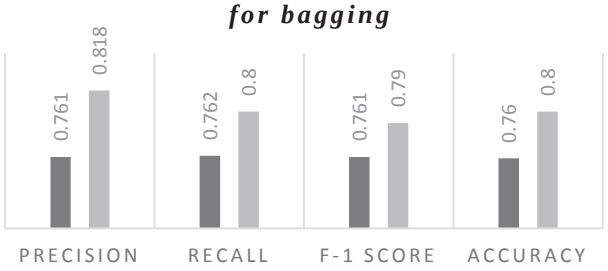

Figure 3: Performance comparison for multi-layer perceptron

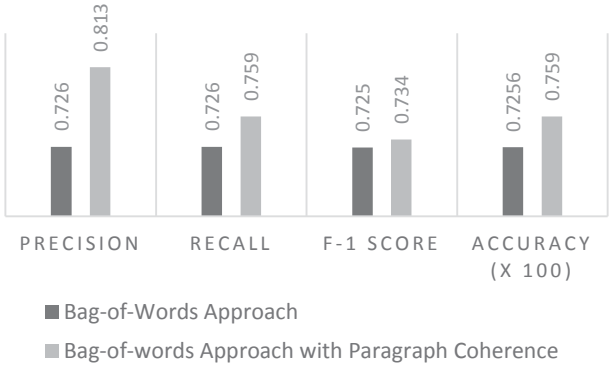

Table II: Classification accuracies

Classifier	Specifications and comments	Accuracy
Multi-layer Perceptron	*Backpropagation algorithm*	75.9%
AdaBoostM1- I	*Base classifiers and their weights:* *Random forest of 10 trees, each constructed while considering 5 random features.*	73.5%
AdaBoostM1- II	*Base classifier: SMO with Polykernel*	77%
Multi-Class Classifier	*Base classifier: Multinomial logistic regression with a ridge estimator* *Method: 1-against-all*	79.5181 %
Bagging	*Base classifier: Multinomial logistic regression with a ridge estimator*	80.7229 %
Multinomial logistic regression with a ridge estimator-I		80.7229 %
Multinomial logistic regression with a ridge estimator-II	*Uses Conjugate Gradient Descent for search for paramenters*	**83.1325%**

Figure. 1, Figure 2 And Figure 3 show performance comparisons for three classification algorithms with the inclusion and exclusion of the Semantic Paragraph Coherence feature. From these figures, it can be easily seen that an approach involving the Semantic Paragraph Coherence feature has superior prediction accuracies than an approach without it. The effectiveness measures reported by the proposed approach are also higher than the basic bag-of-words approach.

5 Conclusion

The contribution of the NLP feature, Semantic Paragraph Coherence together with length of the text cannot be overlooked in the light of the results obtained. That a pure bag of word approach may be limiting and that it should be combined with a knowledge engineering approach is particularly insightful. Classification accuracies that are at par with expert rater-reliability are unheard of in the work done on the automation of integrative complexity scoring. However, the precision, recall and F-1 measure are better performance measures in this body of work, considering the size of the dataset.

Future work could experiment with a much larger dataset. The development of NLP-focused features to aid in the detection of differentiation and integration could not only assist in the scoring of Integrative Complexity, but also help us attain a deeper understanding of human language, and the structure of thought.

6 References

[1] P. Suedfeld, P. E. Tetlock, & S. Streufert, Conceptual/integrative complexity. In C. P. Smith, J. W. Atkinson, D. C. McClelland, and J. Veroff (Eds.), *Motivation and personality: Handbook of thematic content analysis* (pp. 393-400). New York: Cambridge University Press, 1992.

[2] P. E Tetlock, "Cognitive style and political ideology", *Journal of Personality and Social Psychology*, vol. 45, pp. 118-126, 1983.

[3] D. H. Gruenfeld & A. B. Hollingshead, "Sociocognition in work groups: The evolution of group integrative complexity and its relation to task performance", *Small Group Research*, vol. 24, pp. 383-405, 1993

[4] G. J. Feist, "Personality and working style predictors of integrative complexity: A study of scientists' thinking about research and teaching", *Journal of Personality and Social Psychology*, vol. 67, pp. 474-484, 1994.

[5] P. E. Tetlock, "Cognitive style and political belief systems in the British House of Commons", *Journal of Personality and Social Psychology*, vol. 46, pp. 365-375, 1984.

[6] P.E. Tetlock., K. A. Hannum, & P. M. Micheletti, "Stability and change in the complexity of senatorial debate: Testing the cognitive versus rhetorical style hypotheses", *Journal of Personality and Social Psychology*, vol. 46, pp. 979-990, 1984

[7] P. E. Tetlock, & R. Boettger, "Cognitive and rhetorical styles of traditionalist and reformist Soviet politicians: A content analysis study." *Political Psychology*, vol. 10, pp. 209-232, 1989.

[8] Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). Linguistic Inquiry and Word Count: LIWC [Computer software]. Austin, TX: LIWC.net

[9] Y. R. Tausczik & J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods*." Journal of Language and Social Psychology*, vol. 22, pp. 24-54, 2010.

[10] A. K. Ambili, and K. M. Rasheed. "Automated Scoring of the Level of Integrative Complexity from Text Using Machine Learning." In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pp. 300-305. IEEE, 2014.

[11] C. T. Meadow, B. R. Boyce, and D. H. Kraft, *Text Information Retrieval Systems*. 2nd. Ed. Academic Press, 2000

[12] Y. Li, Z. A. Bandar, & D. McLean, "An approach for measuring semantic similarity between words using multiple information sources", *Knowledge and Data Engineering*, IEEE Transactions on, vol. 15(4), pp. 871-882, 2003.

[13] C. Fellbaum, C. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[14] (2010) WordNet: An electronic lexical database.. [Online] Available: http://www. cogsci. princeton. edu/wn

[15] A. G. Miller, "WordNet: a lexical database for English", *Communications of the ACM*, vol. 38(11), pp. 39-41, 1995.

[16] Li, Y., Bandar, Z., McLean, D., & O'Shea, J. (2004). A Method for Measuring Sentence Similarity and its Application to Conversational Agents. In FLAIRS Conference (pp. 820-825).

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann & I. H. Witten, "The WEKA Data Mining Software: An Update"*; SIGKDD Explorations*, vol. 11, Issue 1, 2009.

[18] J. Wielemaker, T. Schrijvers, M. Triska & T. Lager, "Swi-prolog", *Theory and Practice of Logic Programming*, vol. 12(1-2), pp. 67-96, 2012.

[19] Y. Freund & R. E. Schapire, "Experiments with a new boosting algorithm", In *ICML*, vol. 96, pp. 148-156. 1996.