Exploring Creative Concepts in the Nearest Neighborhood using Lexical Ontologies

Prasad Pingali¹, J Jagadeesh¹, Vasudeva Varma¹, Bipin Indurkhyá¹,

¹ LTRC, International Institute of Information Technology, Hyderabad, India
[pvpr, j_jagdeesh, vv, bipin]@iiit.ac.in

Abstract. Conceptual blending is an important area of research for creativity modeling. In this paper we present a creativity model that takes an existing blend and generates new blends using the nearest neighborhood replacements from a lexical ontology. For example “Artificial Intelligence” is a compounded concept comprising of “Artificiality” and “Intelligence” as two sub-concepts. After concept generation, the system comes up with “Artificial Creativity” as one of the generated concepts. This project is part of a larger ongoing project called CAPRICON whose goal is to provide with stimuli for new futuristic concepts. This tool is a creativity enhancement tool that helps users enhance their creative thinking by providing stimuli for new concepts.

1 Introduction

The model presented in this paper is part of a larger ongoing project codenamed CAPRICON. The goal of CAPRICON is to be able to predict futuristic concepts and ideas using the already known concepts. This project is based on the conceptual blending theory. Conceptual Blending is a theory of cognition. According to the theory of Conceptual Blending, elements and vital relations from diverse scenarios are “blended” in a subconscious process. This process is known as Conceptual Blending, and is assumed to be ubiquitous to everyday thought and language. Insights obtained from these blends constitute the products of creative thinking. Gilles Fauconnier and Mark Turner developed the theory of Conceptual Blending. This theory is based on basic ideas advanced by George Lakoff [7].

Generating conceptual blends is a challenging problem in computational creativity. Discussed at length in Fauconnier and Turner [3] and Coulson [2], conceptual blending is a theoretical framework for exploring human information integration. It involves a set of operations for combining dynamic cognitive models in a network of "mental spaces", or partitions or speakers' referential representations. Fauconnier and Turner [3] suggest that a small set of partially compositional processes operate in the creative construction of meaning in analogy, metaphor, counterfactuals, concept combination, and even the comprehension of grammatical constructions. Blending processes depend centrally on projection mapping and dynamic simulation to develop
emergent structure, and to promote novel conceptualizations, involving the generation of inferences, emotional reactions, and rhetorical force.

Figure 1: A Conceptual Blend projection of attributes from Generic Space and Inputs I₁ and I₂.

Figure 1 shows a typical conceptual blend framework described by Fauconnier and Turner [3]. As per this framework, each conceptual blend (for example ‘Artificial-Intelligence’) would get its meaning from a set of attributes. These set of attributes (showed by black dots in Figure 1) are derived from the sub-concepts of the blend (namely ‘Artificiality’ and ‘Intelligence’ in the above mentioned example), which are shown, by Input I₁ and I₂ in the Figure 1. Some of the attributes are common to both I₁ and I₂ and are always projected to the blend. These are known as ‘Generic Space’. From the remaining attributes from I₁ and I₂, only some are projected onto the blend. An interesting aspect of this framework is that it explains the way in which input spaces, which belong to totally diverse domains, can be combined together to arrive at novel and creative meanings with attributes that are not part of either of the input domains.

Fauconnier and Turner have also proposed a set of eight optimality principles as part of this framework. These optimality principles or pressures depict how “good” or “bad” a given blend is. These optimality principles were later formalized by Francisco and Amilcar [14]. Such formalization would help in qualitatively analyzing the conceptual blends. These optimality principles are summarized below.

Integration - The blend must constitute a tightly integrated scene that can be manipulated as a unit. More generally, every space in the blend structure should have integration.
Pattern Completion - Other things being equal, complete elements in the blend using existing integrated patterns as additional inputs. Other things being equal, use a completing frame that has relations that can be the compressed versions of the important outer-space vital relations between the inputs.

Topology - For any input space and any element in that space projected into the blend, it is optimal for the relations of the element in the blend to match the relations of its counterpart.

Maximization of Vital Relations - Other things being equal, maximize the vital relations in the network. In particular, maximize the vital relations in the blended space and reflect them in outer-space vital relations. Turner and Fauconnier identify 15 different vital relations: change, identity, time, space, cause-effect, part-whole, representation, role, analogy, disanalogy, property, similarity, category, intentionality and uniqueness.

Intensification of Vital Relations - Other things being equal, intensify vital relations.

Web - Manipulating the blend as a unit must maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation.

Unpacking - The blend alone must enable the understander to unpack the blend to reconstruct the inputs, the cross-space mapping, the generic space, and the network of connections between all these spaces.

Relevance - Other things being equal, an element in the blend should have relevance, including relevance for establishing links to other spaces and for running the blend. Conversely, an outer-space relation between the inputs that is important for the purpose of the network should have a corresponding compression in the blend.

2 Related Work

Dr. Divago [11] is a free concept generation system, which employed Fauconnier’s framework. The mechanism of bisociation of Divago follows the principle that, when one part of a concept is transferred to another concept, it gets a different meaning. For example, if we transfer the "body" of the concept of woman to the concept of guitar, then the latter gains a different meaning, let us call it a guitar-woman, a bisociation of guitar with woman. Divago uses a computational model of Conceptual Blending [3; 14] to determine which knowledge structures should be transferred at each time. The result is called a blend. For any two concepts, there is an extremely large number of possible blends [13], indeed some of the steps of the conceptual blending are non-deterministic and, for this reason, Divago uses a genetic algorithm to select, from this large search space, the blends that best respect the goal given externally. The usefulness of a generated concept is measured using a fitness function, which is a weighted combination of the eight optimality principles of Fauconnier’s framework.
Therefore in Divago, the goal of the system is to select the best possible interpretation of a given noun-noun combination. The system that we present in this paper is complementary but not competing to the Divago system. Our system focuses on generating multiple blends while Divago works on interpreting such blends. Therefore our system is a complementary system to Divago and not a competing system. One way of looking at this combination is that our system would generate new concepts automatically while Divago would try to generate natural language interpretations of these concepts.

3 Creativity Model using Neighborhood

We present a creativity model in this paper that takes an existing known concept and search its neighborhood in the knowledge base to obtain new concepts. In order to define a neighborhood to a given concept, we need to come up with a similarity measure. We chose the lexical semantic relations as similarity relations to define the neighborhood of a given concept. In our model the input and the output generated are different from Divago, while both the systems deal with Concept blends. The input concept to the system is a compound noun, which is already a known blend (unlike two input spaces as in Divago) and exists in the knowledge base. Since the input concept in our system is already known in the knowledge base, the input concept is already proven to be useful and a creative concept.

In order to generate novel concepts, the system takes a compounded concept as input and generates many more compound concepts based on certain heuristics. The choice of these heuristics is in such a way that theoretically the generated concepts should form “good” blends. For example an input could be 'Machine Learning' and the output of the system could be about a few hundreds of new concepts.

3.1 Concept Generation

To enable the generation of new concepts we make use of lexical ontologies or concept maps. Concept maps are semantic networks wherein concepts are ordered based on different semantic relations among them. A concept map therefore is a graph where each node is a concept and these concepts are connected with each other through arcs. Each arc denotes a type of relation between the two connecting concepts. One such lexical concept map is WordNet [9], which we used in this project.

The system reads the input concept and breaks it down into its constituent words ('Machine Learning' is broken into 'Machine' and 'Learning'). This function can also be treated as 'Unpacking' [14] as described in Fauconnier’s Optimality principles. While unpacking is done, we perform a word sense disambiguation to obtain the most probable senses of the unpacked words. From figure 1, this would mean that the input, which is a blend, is broken down into $I_1$ and $I_2$. Now each of the constituent words in the input concept is replaced by alternative words from the concept map. Thus new concepts are generated. Instead of replacing the alternative words with any other
random word from the concept map, we use certain semantic relations. The intuition behind replacing only semantically related words is that, related words can be considered as the nearest neighbors of the given input concept. Nearest neighbors in a concept map typically have most of the attributes common only differing in minimal number of attributes, hence the generated concept would have a better likelihood of forming a meaningful blend. A pictorial representation of nearest neighborhood concept replacement is shown in Figure 2.

![Figure 2: Nearest Neighborhood Concept Replacement](image)

In Figure 2, the black dots in the middle represent the sub-concepts of input concept (‘Machine’ and ‘Learning’ in case of ‘Machine Learning’, also known as an ‘Anchor Concept’). The nearest neighborhood of the sub-concepts is shown using the dotted circle around the sub-concepts in the Figure2. If we try to interpret these operations from the semantics of the Fauconnier’s Concept Blend model as shown in Figure1, the Input Spaces I₁ and I₂ are replaced with sub-concepts, which are semantically very close to the initial sub-concepts. This implies that, even if we do not know the Generic Space and the attributes that are coming into the blend, intuitively most of the projected attributes in the new blend will remain as the initial projections, except very minute changes. Therefore by performing this operation we did not have to analyze each attribute projection coming from the Generic Space as well as those coming from the Input Spaces I₁ and I₂.

Let us say the input to the system is the blend B.

\[
B \text{ can be defined as a tuple } (I_1, I_2, I_g, \varphi, \sigma, I_b) \text{ where } I_1 \text{ is the Input Domain 1 and } I_2 \text{ is the Input Domain 2. } I_g \text{ and } I_b \text{ are the generic domain and the blend domain respectively, } \varphi \text{ and } \sigma \text{ are a metaphor mapping function and a blending translation operator respectively.}
\]

Further a domain can be defined as a pair of Concept Map and Domain Rules, which constitute the relational and procedural knowledge of the domain. In our concept generation operations we consider only the relational knowledge coming from the
concept map. Therefore if we were to replace one of the input domains with one from its nearest neighbor, let us consider replacing an input domain $I$ with $I'$.

Let $CM=(LC, LR, LCM)$ and $CM'=(LC', LR', LCM')$ be the concept maps related to $I$ and $I'$ respectively, where, $LC, LC'$ (set of concepts) $\in$ language $L$, $LR, LR'$ (set of relations) $\in$ language $L$ and $LCM, LCM'$ are a set of literals of the form $X(Y,Z)$, such that $X \in LR, LR'$ and $Y, Z \in LC, LC'$. We define a distance function $\delta$ between any two concept maps as a function of number of differences in $LC, LR$ and $LCM$. We choose domain space replacements from the neighborhood concepts such that the distance function $\delta$ is minimized.

By the virtue of definition and organization of lexical ontologies, the distance function $\delta$ is directly proportional to the lexical relational distance in the concept map. This feature holds good for various lexical relations such as hypernymy, hyponymy, sibling relations, antonymy [9] etc.

If for each type of relations $i$, the system can find $C_i$ number of replacement concepts, the total number of concepts generated by the system would be $\sum C_i$, provided there are no duplicate concepts for replacements.

For example, given an input concept ‘Machine Learning’, a few of novel concepts generated are shown below.

‘Machine Attention’
‘Machine Inattention’
‘Machine Intention’
‘Machine Intuition’
‘Machine Believing’
‘Machine Perception’
‘Machine Apperception’
‘Machine Classification’
‘Machine Discrimination’
‘Machine Unlearning’
‘Machine Learning Ability’
‘Machine Learning Inability’
‘Machine Learning Disorder’
‘Machine Sleep Learning’

etc.

Therefore the whole operation of understanding what attributes are being projected to the new blend is achieved as a black box in our system. In other words, our system lays more emphasis on focus towards the initial concept, as opposed to total randomness.
4 Experiments and Discussion

In order to evaluate the system, we generated novel concepts taking some already known concepts as inputs to the system. For example if a concept before year X is given as input to the system, we tried to see how many concepts did it predict that were coined after year X. When we tried giving “Artificial Intelligence” as input to the system, the system predicted “Artificial Creativity” as a concept, which came much later than the input concept itself. In order to determine the dates of introduction of concepts, we used a research paper meta-data database downloaded from Citeseer\(^1\). Currently we have tested the system with 25 input concepts and were able to predict 54 concepts, which occurred later than the given input concepts. Some of the example concepts generated are shown in the table below.

<table>
<thead>
<tr>
<th>Input concept</th>
<th>Known concepts generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>inductive logic</td>
<td>analytic logic, deducible logic, inferential logic, causative logic, deductive logic, a_priori logic, a_posteriori logic, noncausative logic, causal logic, inductive propositional_logic, inductive symbolic_logic, inductive boolean_logic</td>
</tr>
<tr>
<td>data mining</td>
<td>art_collection mining, string mining, content mining, corpus mining</td>
</tr>
<tr>
<td>information entropy</td>
<td>information bandwidth, information length, information randomness, information invisibility</td>
</tr>
<tr>
<td>Input concept</td>
<td>Known concepts generated</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>computer vision</td>
<td>computer perception</td>
</tr>
<tr>
<td></td>
<td>computer sound</td>
</tr>
<tr>
<td></td>
<td>computer hearing</td>
</tr>
<tr>
<td></td>
<td>computer creativity</td>
</tr>
<tr>
<td></td>
<td>computer inventiveness</td>
</tr>
<tr>
<td>software engineering</td>
<td>hardware engineering</td>
</tr>
<tr>
<td></td>
<td>computer engineering</td>
</tr>
<tr>
<td></td>
<td>application engineering</td>
</tr>
<tr>
<td></td>
<td>database_management_system</td>
</tr>
<tr>
<td></td>
<td>engineering</td>
</tr>
<tr>
<td></td>
<td>software plan</td>
</tr>
<tr>
<td></td>
<td>software artificial_intelligence</td>
</tr>
<tr>
<td></td>
<td>software communications</td>
</tr>
<tr>
<td></td>
<td>software biotechnology</td>
</tr>
<tr>
<td></td>
<td>software design</td>
</tr>
<tr>
<td></td>
<td>software architecture</td>
</tr>
<tr>
<td></td>
<td>software project</td>
</tr>
<tr>
<td>speech processing</td>
<td>slang processing</td>
</tr>
<tr>
<td></td>
<td>spell processing</td>
</tr>
<tr>
<td></td>
<td>pronunciation processing</td>
</tr>
<tr>
<td></td>
<td>idiom processing</td>
</tr>
<tr>
<td></td>
<td>dialect processing</td>
</tr>
<tr>
<td></td>
<td>prosody processing</td>
</tr>
<tr>
<td></td>
<td>language processing</td>
</tr>
<tr>
<td></td>
<td>text processing</td>
</tr>
<tr>
<td></td>
<td>intonation processing</td>
</tr>
<tr>
<td></td>
<td>ambiguity processing</td>
</tr>
<tr>
<td></td>
<td>speech data_mining</td>
</tr>
<tr>
<td></td>
<td>speech transformation</td>
</tr>
</tbody>
</table>
A lot of other concepts were generated in the process, of which many could be coined in future. Since the system generates a huge number of concepts for a single given input concept, we are currently working on a concept ranking function. The goal of this function would be rank the generated concepts in descending order of usefulness and novelty.

5 Conclusion and Future Work

In this paper we showed that creative and novel ideas can be generated by the machine in the form of concept compounds using an input anchor concept and its neighborhood in a lexical ontology. Also each concept generated could have more than one interpretation. However we did not attempt to analyze the meaning of each of the generate concepts automatically as Divago does. Our system is a complementary system to Divago and not a competing system. In our approach we also see that we depend on the neighborhood of the input concept, which is based on an existing ontology. While this is a good approach to focus the generated concepts very close to the input concept, we think that cross-domain blends, which could also be sometimes interesting, may not be collected. We would like to introduce some mechanism to blend distant concepts as well into our system as part of our future work. Since this might become computationally expensive, we would like to find strategies to minimize expensive computations. Since the number of generated concepts can be huge, we plan to introduce a ranking function, which can rank these concepts based on a usefulness function.

In future we would also like to identify and use more semantic relations between concepts to generate novel concepts. One such relation could be finding analogies through an analogy search for a given concept. For example a “Jet Plane wing” as input can yield “Albatross wing” as an analogy, since both of them fly long distance, and are designed to minimize drag and so on. Instead of using deductive logic to find such analogies, we plan to use text-mining techniques from large corpora.

References