ARCHITECTURE FOR AUTOMATIC POETRY GENERATION THROUGH PATTERN RECOGNITION

by

Kimberly Scofield

B.Sc., University of Northern British Columbia, 2011

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
INTERDISCIPLINARY STUDIES

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

April 2017

Abstract

Document representation and topic modelling are important problems for artificial intelligence researchers, with applications ranging from education technology to bioinformatics. Many approaches have been proposed, the majority falling broadly into categories of Statistical Analysis and Natural Language Processing (NLP). This thesis proposes an architecture that optimizes a combination of statistical and linguistic analysis in an unsupervised machine learning environment.

The proposed architecture is a design for agile, stable, document modelling. By clustering within the statistical inference algorithm, it reduces the computational cost of time and space associated with conventional classifying algorithms such as K-means, increasing the threshold for size and frequency of aggregate data analysis. This translates to an increased stability for evolution of learning. The architecture builds on the concept of socio-linguistic connections as an inherent combination of statistics and linguistics, and employs well-researched concepts of statistical and linguistic analysis, including embedded sub-manifold analysis. It optimizes both linguistic connections and computational cost.

Trials are run with three sets of parameters, and results distributed for volunteer evaluation. Feedback from the evaluation indicates that the proposed architecture produces groups of sentences (poems) with a high degree of social acceptance and response.

Contents

List of Figures

List of Algorithms

List of Tables

Acknowledgement

First, and most important, I wish to acknowledge the support and encouragement I have received from my family while on this journey. Their belief in me and their unfailing support has been invaluable. I also want to thank my friends for their support and encouragement through everything, and for sticking around even when my time with them was limited. I couldn't have done this without all of you - thank you for never letting me doubt myself!

Next, I wish to acknowledge the guidance, persistence, and encouragement of my co-supervisor, Dr. Liang Chen. His belief in my project has been both unfailing and encouraging; his belief in me has been inspiring. Thank you also to my co-supervisor Dr Charles Brown, whose love of linguistics and belief in me was the start of my journey, and whose guidance showed me which parts of the field I love most. Thank you both for everything you have done for me over the last few years.

Chapter 1

Introduction

1.1 Overview

This chapter provides an overview of the thesis topic, as well as the proposed architecture and methodology. Document representation and topic modelling are key components in finding relevant, meaningful, linguistic relationships within a text. The purpose of this project is to look at these relationships from a perspective different than the conventional document-term joint-probability analysis of a large corpus, and focus instead on the advantages to be gained from analysis and modelling with smaller "documents" (sentences, in this case), sociolinguistic feature sets, and statistical analysis that goes beyond Euclidean space. The result is an environment that optimizes the discovery of linguistic connections, while reducing the computational cost of space and time.

Methodology

- Linguistics and semantics are defined within a relaxed structure. There is no conventional part-of-speech-tagging, and only broad structural categories to

guide word placement and choice.

- The native dictionary can be adapted to target specific baselines, such as reading level.

- The statistical / Natural Language Processing (NLP[1]) order of analysis, and the specific methods used, have been carefully chosen to preserve and promote linguistic meaning while at the same time minimizing matrix size and sparsity.

- Conventional clustering is not used: clustering is accomplished by the statistical algorithms in the embedded submanifold code, reducing computational cost (space and time) and allowing an increased threshold for stable evolution of learning.

Architecture

- Pluggable architecture: the customizable semantic template and dictionary can accommodate different languages and reading levels.

- Block style components: each section, or block, executes one discrete part of the process, allowing changes to be made and the effects tracked.

- Adjustable parameters: a native toolkit to integrate block functionality allows coarse- and fine-grained optimization at many levels, to adjust output for different ages, abilities, and output goals.

Why Poetry?

Poetry offers a viable model of the human creative process, linking thought process to emotive expression. This is the crucial area of information selection: identifying

[1]Natural Language Processing

the data and parameters that most efficiently and accurately bridge a humanized extrapolation from data, to information, and finally to knowledge.

Free form poetry does this in a landscape of (comparatively) relaxed syntactic structure, allowing expression, abstraction and extrapolation to be of higher priority than form. As Flores states in [21] *Understanding Computers and Cognition: A New Foundation for Design*: "one of the most prominent illusions... is the belief that knowledge consists of formal theories that can be systematically used to make predictions." The relaxed structure of free form poetry allows the algorithm to create a series of abstract associations, building on the inherent cultural and linguistic markers present in the sentence structure and the dictionary. The goal is to emulate not the process of human thought but the result.

1.2 Contribution

The main contribution of this thesis is a design for unsupervised, agile, stable document modelling, simulated within an architecture for the automatic generation of poetry. This architecture combines well-researched methods of statistical and NLP analysis in a structure of independent blocks. Each block uses customized parameters specifically chosen to optimize extraction of information that is culturally and socially relevant, as well as optimizing computational cost.

The goal of the project is to create a framework capable of generating poetry that will pass as human-created. To accomplish this goal, the proposed architecture builds on the concept of sociolinguistic connections as a natural combination of statistical probability, linguistics, and relevant vocabulary.

The architecture creates simulation data using a pseudo-random generated set of sentences and a custom dictionary. Dimensionality reduction is done first, to

preserve inherent and/or hidden semantic integrity while at the same time reducing noise and sparsity. This is followed by statistical analysis within an embedded submanifold model [2]. Linguistic analysis techniques are then used to cluster groups of semantically related sentences.

This architecture was validated in the context of a modified Turing Test: volunteers read and evaluated a a set of unlabelled poems, created by both human poets and the Automatic Generator. The results show that the architecture produces sentences and groups of sentences (poems) with a high degree of social acceptance, concept fluidity, and information extrapolation. Of the varied number of clusters examined, trials using 5 clusters were found to be the most effective. These were run on files of 3000-3500 sentences, each sentence consisting of between 3 and 12 words.

The organization of the remaining sections is as follows: Chapter 2 covers background and related work. Chapter 3 is a general overview of the proposed architecture and its methodology. Chapter 4 is a detailed look at each of the architecture's components/blocks. Chapter 5 provides experimental results and analysis. Chapter 6 concludes the thesis with a summary of the concepts presented, and some thoughts and ideas for future work.

Chapter 2

Background and Related Work

This chapter presents background information and an overview of previous work in document representation and topic modelling, statistical and NLP[1] analysis, and sociocultural linguistics.

Document representation and topic modelling encompass broad fields of application, and many different implementation strategies within these fields. For the purpose of this project we focus on document representation as a means of identifying relevant structures within a text document (punctuation, words, document begin/end points, etc) and topic modelling as the clustering of conceptually connected words, and groups of words, across all documents within a corpus.

For researchers in this field the question has always been: how to create a robust, self-evolving model that can dynamically adapt to random user input while producing a relative, pragmatic output aggregated across all data, both historical and current.

[1]Natural Language Processing

2.1 Linguistic Integrity

It is important to note that this project has no interest in teaching computers to use language as a human would use it. Rather, this project is interested in a method to draw conclusions from human communication and to extrapolate these conclusions. Linguistic integrity addresses the creation of appropriate, pseudo-random, input data that is carefully structured to simulate a reasonable baseline learning environment. For this we have precedent from both the computational and linguistics community:

> [12] George Luger, *Artificial Intelligence*:
>
> "Digital computers are not merely a vehicle for testing theories of intelligence. Their architecture also suggests a specific paradigm for such theories: intelligence is a form of information processing. The notion of a problem-solving methodology, for example, owes more to the sequential nature of computer operation than it does to any biological model of intelligence." (p 12)

> [21] Fernando Flores, *Understanding Computers and Cognition*:
>
> "computers will remain incapable of using language in the way human beings do, both in interpretation and in the generation of commitment that is central to language." (p 12)

> [9] Dan Jurafsky and James Martin, *Speech and Language Processing*:
>
> "The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities (p 13, from Zellig Harris, 1968)"

In other words... "You shall know a word by the company it keeps." [15] (Ted Pedersen's paraphrase of Harris, 1954 and Firth, 1957)

Maximizing the retention and visibility of the underlying meaning that is inherent in the data means creating simulation data with a balance between a complete lack of syntactic structure (no word order, gibberish) and a basic word order, as could conceivably be acquired when someone learns a new language by listening to others speak. A computer could arguably learn these same basic word placements by analyzing word patterns in documents - in effect "learning" rudimentary word placement. This new-language-learner semantic structure has been simulated with a custom dictionary and a modified Sci-Gen [20] document template.

On these concepts both sides of the "nature versus nurture" crowd can play nicely together in some instances: "theories of cognition can deal purely with 'competence', characterizing the behavior of HCI cognitive systems while making no hypothesis concerning the generation of that behavior by mechanisms." (Noam Chomsky, as cited in [21] *Understanding Computers and Cognition: A New Foundation for Design*).

Linguistics is the scientific study of language and its structure, including the study of morphology, syntax, phonetics, and semantics. Computational linguistics is that branch of linguistics in which the techniques of computer science are applied to the analysis of language. A sociolinguistic approach, in combination with the new-language-learner vocabulary and semantic structure, allows the AutoGen architecture to see not only valid, but also culturally relevant, patterns (concept connections) in the text.

As humans, each of us has access to a vast internal storehouse of subtle, subconscious, language-related knowledge: a synthesis of experience, social interaction, and personal reflection. Linguistics is a tool that we can use to tap into this constantly evolving knowledge base.

In [21] *Understanding Computers and Cognition* Flores states: "language, and

therefore thought, is ultimately based on social interaction." Social interaction is the vital link that transforms information into usable knowledge, knowledge that is both socially and culturally relevant. Supervised machine learning (training sets that determine weights) is one way to simulate a baseline for social interaction.

This project uses unsupervised machine learning and looks at social interaction from a different perspective: the perspective of patterns.

Patterns are the basis of computational linguistics: recurrence equates to patterns. Computers do well as "devices for facilitating human communication in language... [they] observe and describe regular recurrences" [21]. Therefore, linguistic meaning can be related to language structure as pattern and pattern recurrence.

The proposed architecture does not link speech act and listener, it links a pattern of speech act recurrence, or a "pattern of acts through time." [21] and by using the embedded submanifold protocol of [2] Deng Cai et al, we can bring visibility to 3+ dimensions of contextual meaning.

New studies show there is a high likelihood that humans do not store knowledge, that we create, instead, well-used neural pathways that function as 'memory recall'. By this theory, a 'right decision', such as recalling our route from home to work, is no more than the execution of a well-used neural path, or a neural path of least resistance [4]. Identifying conceptual patterns in text can also be seen to result from repetition (patterns) rather than accumulated knowledge. Echoing the earlier quote from both Harris and Firth, we can see that to know a word by the company it keeps (by its associations, placement, and frequency of occurrence) is equivalent to recognizing its pattern of recurrence.

Shared assumptions and cultural anomalies also become less relevant when we add:

> "focus should be on the interactions within the system as a whole, not

> on the structure of perturbations. The perturbations do not determine what happens in the nervous system, but merely trigger changes of state. "interactions and transformations continuously regenerate the network of processes (relations) that produced them". [21] (Flores and Winograd, (quoted from: Maturana and Varela, *Autopoiesis and Cognition*, 1980, p 19))

And:

> We consciously register the results of our understanding and thinking ... but not the understanding and thinking processes themselves; and these symbolic abstractions, to the extent that they lack quantitative or probabilistic dimensions, can lead us to suppose that the underlying processing is nonquantitative as well. But the successes of statistical NLP[2], as well as recent developments in cognitive science (e.g., Fine et al. 2013; Tenenbaum et al. 2011; Chater and Oaksford 2008) suggest that language and thinking are not only symbolic, but deeply quantitative and in particular probabilistic. [19] (from *The Stanford Encyclopedia of Philosophy*)

In other words successful simulation of human thought relies more on understanding **what** is happening, than on **why** it is happening.

2.2 Statistical Processing Algorithms

2.2.1 VSM: Vector Space Model

VSM creates a document-term matrix with each document represented as a "bag of words", or sparse term-frequency vector, in which the order of the words is not important [17]:

[2]Natural Language Processing

The student runs an event.

Students study running.

The student's car runs fast.

Table 2.1: First Order Vectors of Unigrams

	stud	run	event	car	fast
sent-1	1	1	1	0	0
sent-2	2	1	0	0	0
sent-3	1	1	0	1	1

Table 2.1 shows a sparse first order co-occurrence matrix with each document (row) viewed as a term-frequency vector.

A common method for representing the similarity of documents is the cosine similarity measurement: the cosine of the angle between two document vectors is calculated and similarity is expressed as the normalized inner product of this calculation:

$$sim(d_1, d_2) = \frac{\mathbf{V}(d_1) \cdot \mathbf{V}(d_2)}{|\mathbf{V}(d_1)||\mathbf{V}(d_2)|} \tag{2.1}$$

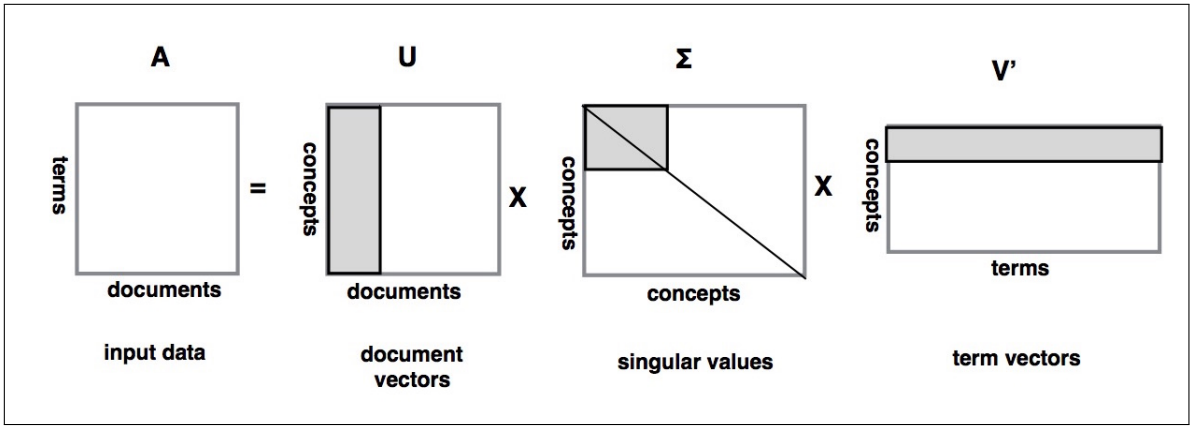
VSM has severe drawbacks, due mainly to the ambiguity of words (polysemy), individual differences in word usage (synonymy), and variations in personal writing style. VSM simply counts the number of occurrences of each word and therefore does not capture subtle differences in meaning based on context; for example, " a lean person" and "a leaning building".

2.2.2 LSI: Latent Semantic Indexing

Latent Semantic Indexing (LSI), also called Latent Semantic Analysis (LSA), was first proposed by Bellcore Laboratories in the late 1980's [3].

LSI was developed to improve VSM performance and uses Singular Value Decomposition (SVD) to reduce the dimensionality of the document-term matrix, and increase visibility of the documents' latent semantic subspace [8]. This process of matrix reduction includes the amalgamation of similar words and word combinations, effectively reducing the ambiguity prevalent in the VSM model.

Figure 2.1: Singular Value Decomposition (SVD)



Statistically, the n-highest singular values from matrix A can be used to produce the best n-rank representation of the data with the equation:

$$A = U\Sigma V_T \tag{2.2}$$

SVD performs best with normally-distributed data, as would be represented by a square "Matrix A" but, in general, a natural language matrix is not square and

[3]G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The Vocabulary Problem in Human-System Communication: an Analysis and a Solution, Bell Communications Research, 1987

its data is not normally distributed. The SVD algorithm used in LSI has been generalized to accept input data from a non-square matrix but it comes with a caveat: the LSI model itself works best when used to identify semantic similarity between documents that do not appear, on the surface, to be similar. LSI performance is degraded if documents are homogeneous (similar language, dialect, subject matter, vocabulary), as is common in many natural language settings [14].

SenseClusters is an example of an LSI model that uses SVD for concept clustering. It is a graduate project created by Professor Tom Pedersen and his students at the University of Minnesota: "SenseClusters was implemented at the University of Minnesota, Duluth by Amruta Purandare (2002-2004) and Anagha Kulkarni (2004-2006), with support for Latent Semantic Analysis being added by Mahesh Joshi in the summer of 2006."[4] [15]. SenseClusters uses native methods and LSI to cluster sentences in a semi-supervised machine learning environment. It uses WordNet[5] definitions for weights and disambiguation, and a tagged benchmark corpus for testing.

2.2.3 PLSI: Probabilistic Latent Semantic Indexing

PLSI is a generative probabilistic model used to maximize the joint probability of documents and words in a corpus. PLSI does this by estimating the probability distribution for each document independently; therefore, the number of parameters increases linearly with the number of documents. This leads to overfitting: an excess of parameters relative to the number of documents. This noisy environment can exaggerate minor fluctuations in the data and obscure concept connections.

Below is the PLSI algorithm pseudocode from the paper "Modeling Hidden Top-

[4]http://senseclusters.sourceforge.net/

[5]tagged online dictionary, http://wordnet.princeton.edu/

ics on Document Manifold" by [2] Deng Cai et al, showing the modification in step 2 that is the key to avoiding overfitting in Algorithm 2.1[6]:

[6] $\propto$: proportional to

Algorithm 2.1 Generalized EM Algorithm for PLSI

Variables:

N = number of documents

K = number of latent topics

M = number of terms (words) in the dictionary

Steps:

1. select a document d_i with probability $P(d_i)$
2. pick a latent topic z_k with probability $P(z_k|d_i)$
3. generate a word w_j with probability $P(w_j|z_k)$

Result:

observation pair (d_i, w_j) (latent topic variable z_k is discarded)

Joint probability representation:

$$\begin{aligned} P(d_i, w_j) &= P(d_i)(w_j|d_i) \\ P(w_j, d_i) &= \Sigma_{k=1}^{K} P(w_j|z_k)P(z_k, d_i) \end{aligned} \tag{2.3}$$

Estimate the parameters: maximize the log-likelihood L:

$$\begin{aligned} L &= \Sigma_{i=1}^{N}\Sigma_{j=1}^{M} n(d_i, w_j)\ logP(d_i, w_j) \\ &\propto \Sigma_{i=1}^{N}\Sigma_{j=1}^{M} n(d_i, w_j)\ log\Sigma_{k=1}^{K} P(w_j|z_k)P(z_k|d_i) \end{aligned} \tag{2.4}$$

This results in $NK + MK$ parameters of the form:

$$P(w_j|z_k), P(z_k|d_i)$$

which are independently estimated in the PLSI model. The number of parameters in PLSI therefore grows linearly with the number of training documents (N), and leads to overfitting.

2.2.4 LDA: Latent Dirichlet Allocation

A Dirichlet distribution is a Probability Distribution Function,(P_D) formulated to be a 'measure of measures'. LDA avoids overfitting by modelling the probability distribution of each document, over topics, as a random variable K, where K is the number of hidden topics. Each k in K is a discrete Probability Distribution Function, used to calculate $\alpha_i - \alpha_k$ and α_K is the master P_D for K.

Below is the LDA algorithm pseudocode from the paper "Modeling Hidden Topics on Document Manifold" by [2] Deng Cai et al, showing the modification in step 2 that is the key to avoiding overfitting in Algorithm 2.1[7]:

Algorithm 2.2 Generalized EM Algorithm: Modified Steps for LDA

Steps:

1. select a document d_i with probability $P(d_i)$
2. pick a latent topic z_k
   - 2.1 generate $\theta_i \backsim Dir(\alpha)$
   - 2.2 pick a latent topic z_k with probability $P(z_k|\theta_i)$
3. generate a word w_j with probability $P(w_j|z_k)$

After step 3:

Joint probability representation: Equation 2.3 (PLSI)

Parameter estimation: maximize the log-likelihood Equation 2.4 (PLSI)

This results in $K + MK$ parameters of the form:

$$P(w_j|z_k)$$

The number of parameters, therefore, does not grow linearly with the number of

[7] $\backsim$: distributed according to the distribution

documents and overfitting is not an issue in LDA.

Both PLSI and LDA discover the hidden topics in the Euclidean space. However, Euclidean space is flat and recent studies suggest that a non-linear, low-dimensional manifold, embedded in high-dimensional ambient space [10][23], is a more accurate visualization of a document space. Exploitation of the local geometric structure is essential to reveal the hidden semantics in this high-dimensional space [2] and neither PLSI nor LDA exploit the geometric structure of the document representation. A manifold, then, is a logical next step.

2.2.5 LapPLSI: Laplacian Probabilistic Latent Semantic Indexing

"LapPLSI models the document space as a submanifold embedded in the ambient space and directly performs the topic modelling on this document manifold in question." [2]. This model is only relevant and useful because there exists an identifiable relation (a connection) between:

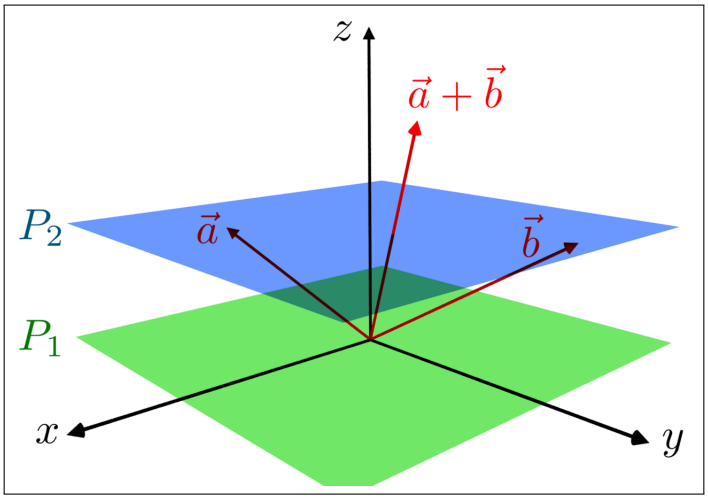
$$P_D \text{ and the Conditional Probability Distribution } P(z|d)$$

This connection is explained below in "The Manifold Assumption", and allows LapPLSI to build on the LDA concept of creating a 'measure of measures', or a 'distribution of distributions', and thereby discover the intrinsic geometrical structure of the document space.

Embedded submanifold: a quick visual: Simplistically, a manifold exists in an affine space, a space that generalizes properties of Euclidean space that are independent of the measurement of distance and angles: in the graphic below, P_2 is *not* a vector subspace of $\mathbb{R}_3$, but it *is* a linear substructure from which relative

measurements of vectors a and b can be made.[8]

Figure 2.2: Affine Space example: $\mathbb{R}eal\ numbers$ in 3 dimensions



To discover the intrinsic geometry, LapPLSI uses a "geometrically based regularizer" (new variable created for the LapPLSI algorithm) and an assumption that there is a relevant, useful, and identifiable relation between the overarching probability distribution P_D and the conditional probability distribution(s) $P(z|d)$ (the probability of specific words). In other words, a relation that connects the ambient space and the submanifold. This is an known as the Manifold Assumption. The regularizer and Manifold Assumption are explained below.

The Geometric Regularizer is a variable that generalizes the Expectation-Maximization (EM) step of LDA (Algorithm 2.2) to 3+ dimensions. The maximum log-likelihood step (Equation 2.4) becomes a regularized log-likelihood ξ:

[8]Graphic by Jan Krieg, https://commons.wikimedia.org/w/index.php?curid=45756454, CC BY-SA 4.0

$$\begin{aligned} \xi &= L - \lambda R = L - \lambda \Sigma_{k=1}^{K} R^k \\ &\propto \Sigma_{i=1}^{N} \Sigma_{j=1}^{M} n(d_i, w_j)\ log \Sigma_{k=1}^{K} P(w_j|z_k) P(z_k|d_i) \\ &\quad -\frac{\lambda}{2} \Sigma_{k=1}^{K} \Sigma_{j=1}^{N} (P(z_k|d_i) - P(z_k|d_i))^2 W_{ij} \end{aligned} \tag{2.5}$$

where λ is the Geometric Regularization parameter

This results in $NK + MK$ parameters of the form:

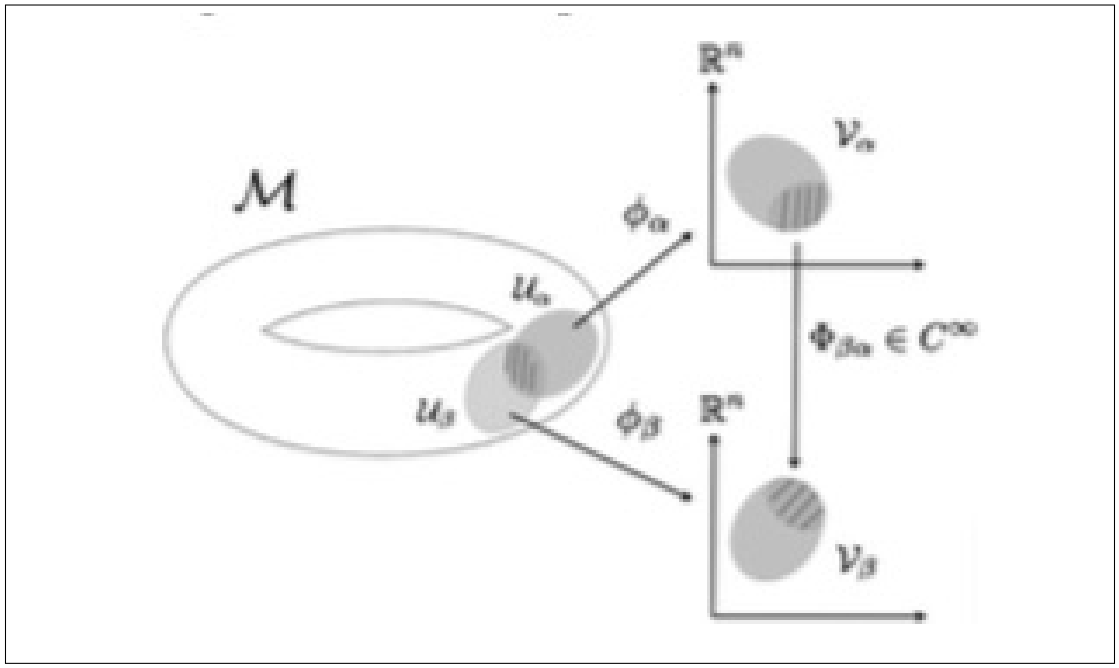
$$P(w_j|z_k), P(z_k|d_i)$$

Manifold assumption: is a method of generalizing vector similarity measurements to 3+ dimensions. In the same vein as the LDA assumptions outlined in Section 2.2.4 in the manifold assumption we assume:

IF: two documents $d_1, d_2 \in D$ are close in the intrinsic geometry of P_D
THEN: their conditional probability distributions $P(z|d_1)$ and $P(z|d_2)$ are similar

In other words, we assume that the conditional probability distribution $P(z|d)$ (probability of latent variable z, given word d) varies smoothly along the geodesics in the intrinsic geometry of P_D.

Figure 2.3: Manifold geodisics: a visual representation of intrinsic geometry



This assumption is used extensively in dimensionality reduction algorithms and semi-supervised learning algorithms [5][7].

Deng Cai et al tested their LapPLSI model using the benchmark Reuters[9] and TDT2[10] databases. Results were validated using similarity measurements that compared the topic labels and document classification of the LapPLSI results to the hand-tagged benchmark labels of each dataset. The Manifold Assumption, using equations from [2] Deng Cai et al:

[9]Reuters corpora is a database of manually categorized newswire stories: http://www.daviddlewis.com/resources/testcollections/reuters21578/

[10]TDT2: Topic Detection and Tracking Evaluation, a database of manually categorized, "multiple sources of information in the form of both text and speech from newswire and radio and television news broadcast programs", http://www.itl.nist.gov/iad/mig/tests/tdt/1998/

Algorithm 2.3 Manifold Assumption Overview

Let: $f_k(d) = P(z_k|d)$ be the Conditional Probability Distribution Function (P_D)

And let: $\|f_k\|_M^2$ be used to measure the smoothness of f_k along the geodisics of the intrinsic geometry of P_D.

Then: the support of P_D is a compact submanifold $M \subset \mathbb{R}^M$ and:

$$\|f_k\|_M^2 = \int_{d \varepsilon M} \| \nabla_M f_k \|^2 \, dP_D(d) \tag{2.6}$$

where: ∇_M *is the gradient of* f_k *along manifold* M

Simplistically, a compact manifold refers specifically to a manifold that is compact on a topological space, and generally implies that the manifold is without boundary. A circle is the only 1 dimensional compact manifold and a sphere is an example of a 2-dimensional compact manifold. Using these examples it is easy to visualize the usefulness of the compact *submanifold* in the LapPLSI algorithm: it can be covered by a finite number of coordinate charts (it can be mapped) and any continuous real-valued function applied to it is bounded. Since the document manifold is generally not known, $f_k(d)$ can't actually be computed, but it can be discretely approximated using nearest neighbour models [2].

Which leads naturally to the topic of clustering...

2.3 Similarity Measurements for Clustering

Classification versus clustering: Classification is a procedure used in supervised learning; clustering is a procedure used in unsupervised learning.

In classification, documents are 'classified' (assigned to a 'class') by hand. For example, each document in the Reuters corpora has been hand-tagged by assigning

to it the topic (label) with which it is most closely aligned: 'news', 'entertainment', etc.

Clustering does not compare each document to a pre-determined set of hand-tagged labels, instead it chooses the most likely cluster, or label, for each document, essentially creating clusters (groups of related documents) in which the documents in each group are more similar to each other than to the documents in other groups. What constitutes this 'similarity' is specific to each application.

Computational cost (efficiency versus accuracy) must be considered in the choice of clustering method, as well as the general shape of the data (natural language generally produces elliptical clusters). Each clustering method is suited to specific data distribution patterns, or shapes, and carries trade-offs in accuracy as well as computational time and space requirements. This project uses unsupervised learning and therefore uses clustering.

Stable evolution of learning: Clustering an aggregate of all data, new and old, each time the algorithm is run, allows results to be created with no preconceived idea (no prior knowledge) of past cluster topics and past members of those clusters. Results obtained in this manner are more accurate than results obtained by running only new data and aggregating with previous results [18]. By running a fast algorithm like LapPLSI, data can be aggregated and re-analyzed more frequently.

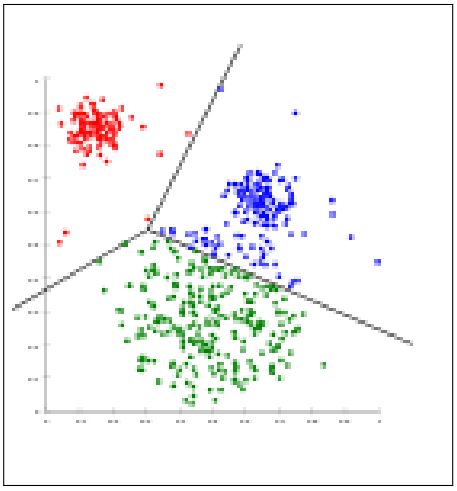
2.3.1 Centroid-based Clustering

In centroid-based clustering, clusters are represented by a central vector (an average of all vectors within the cluster).

The K-means model is an example of centroid-based clustering in which data is partitioned into a Voroni diagram (a visual representation of the borders between

clusters), with fuzzy borders between clusters[11]:

Figure 2.4: K-Means clustering: Voroni Partitioning



In this model objects are assigned to the nearest cluster center, not the nearest cluster, and this results in the optimization of distance-to-cluster-centers, not distance-to-clusters.

K-means is an NP-hard optimization problem that finds only the local maximum; it is also run with either random or pre-assigned cluster centres and there is no guarantee that the number of clusters created, or the centres used in each run, are optimal. This means that each run of the K-means algorithm assigns each document to its most relevant cluster, but during multiple runs of the algorithm documents can shift between clusters. To combat sub-optimal partitioning, K-means is generally run multiple times (10x is common) and the results can be either filtered with fitness criteria or aggregated.

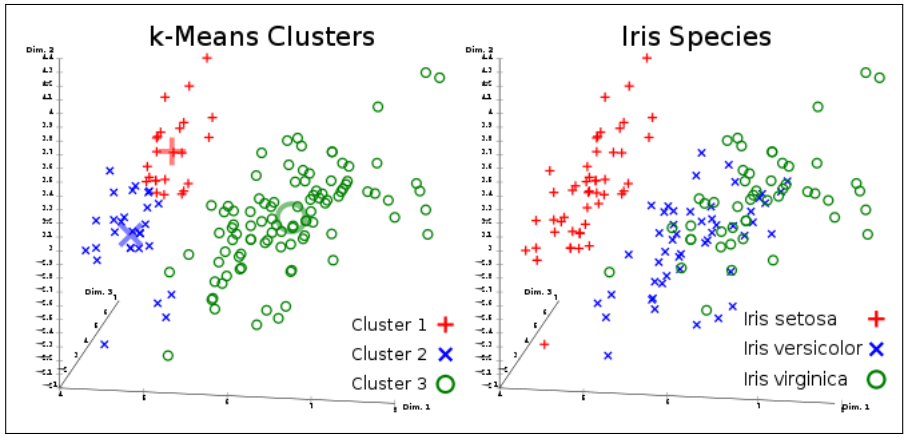
K-means is a fast, versatile clustering method that pulls non-spherical data into spherical clusters, as shown below with the Iris Dataset[12]. It does not create a

[11]Voroni partition graphic: Chire, Own work, https://commons.wikimedia.org/w/index.php?curid=17085714

[12]Fisher, 1936: https://archive.ics.uci.edu/ml/datasets/Iris, "data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other".

probability distribution to allow granularity of clustering based on threshold cluster membership, or analysis based on strength of cluster membership. such as clustering all documents with a 0.90 probability of membership in a particular cluster.[13]

Figure 2.5: K-Means clustering of the Iris Dataset



2.3.2 Distribution-based clustering

Distribution-based clustering is used by statistical analysis algorithms such as LSA, LDA, and LapPLSI. As is shown in the LapPLSI model, distribution-based clustering is able to bring out complex correlations within the data and this ability is optimized when over-fitting is under control.

Distribution-based clustering produces a Conditional Probability Distribution Function P_D consisting of calculations for each word:

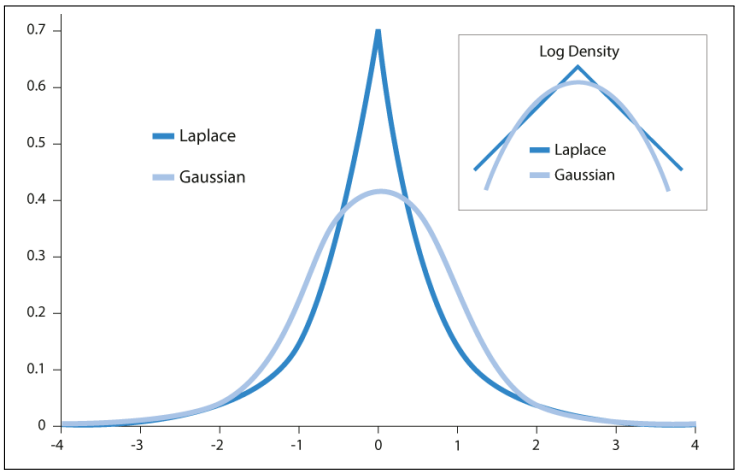
1. a posterior probability $P_(z_k|d_i, w_j)$: the probability of latent variable z, given word w in document d.

2. and a-priori probability: $P_(z_k|d_i)$: the estimate (using the posterior probability) of the probability of latent variable z, given document d.

[13]Iris Dataset graphic: https://en.wikipedia.org/wiki/K-means_clustering

This allows for granular adjustment of such things as cluster shape.

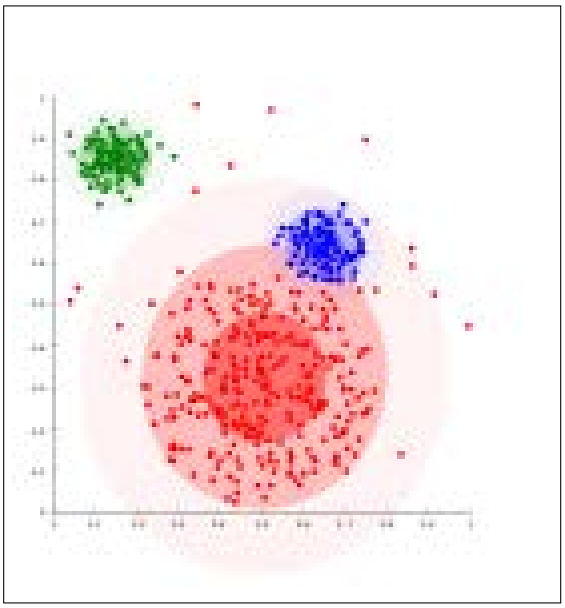
Figure 2.6: Normal Distribution: LapPLSI versus Gaussian



Gaussian mixture models are a well-known example of distribution-based clustering. They are more stable and accurate than the K-means model but less efficient. They work best on data with a compact shape that clusters naturally to centroid shaped clusters; natural language tends to elliptical clusters. To avoid over-fitting, the data is usually modelled with a random initialization of a fixed number of distributions.[14] [1]

Figure 2.7: Cluster: Gaussian example

[14]LapPLSI vs Gaussian Distribution graphic: http://www.europeanfinancialreview.com/?p=217#!prettyPhoto/0/



LapPLSI uses the Laplachian model and customized parameters, as covered in Section 2.2.5. The hidden topics extracted by topic modelling are the clusters and the estimated conditional Probability Distribution Function $P(z_k|d_i)$ can be used as the label of each cluster.

This method produces clusters that are more compact and focused than those produced by more conventional clustering algorithms (for example K-means models), better capturing the hidden geometric structure of the document space [2].

Pragmatically, this translates into the algorithm's ability to find conceptual connections within small documents, especially important in this project, where each 'document' is actually a 'sentence'.

2.4 Natural Language Processing Algorithms

Natural Language Processing (NLP) is a field that explores the interaction of computers with human (natural) languages. For our purposes we will describe NLP

similarity measurements in terms of first and second order matrix representation.[15] The NLP sections of the automatic generator architecture take advantage of the power of a-priori association as bigrams[16] [17].

First-order representation

Firat order representation works with frequency: how many times a word, or group of words, appears in a specific context (sentence, document, etc). If two sentences have a high percentage of words in common, first order methods will rate them as conceptually similar. Since it is likely that these sentences are, in fact, conceptually similar, these methods return fewer false positives than second order methods. Conversely, short sentences that are conceptually similar do not always share a high percentage of words in common, making false negatives an issue for these methods [15]. An example of a first order co-occurrence representation is shown below: a document-term matrix where each column is a term (in this case a bigram) and each row is a sentence. The matrix represents the number of occurrences of each bigram in each sentence:

Example sentences:

- "The happy child and the big dog found a mud puddle."
- "The big dog found the best mud puddle."
- "The orange cat loves raisin toast and sunny places."
- "The big dog chased the orange cat into the lake."

[15] http://www.nltk.org/book/: text& tutorial for NLTK and Python, opensource
[16] n-grams: bigrams, trigrams, etc are groups of words that appear together in a corpus

Table 2.2: First Order Co-occurrence Matrix: vectors of bigrams

	happy child	big dog	raisin toast	orange cat	mud puddle
sent-1	1	1	0	0	1
sent-2	0	1	0	0	1
sent-3	0	0	1	1	0
sent-4	0	1	1	0	0

Second order representation

Second order representation addresses the problem of false negatives. It detects similarity in short sentences by weighting each word according to its importance across the entire corpus:

Step 1: Using the example sentences from the previous section, we create a word by word matrix of bigrams and use Singular Value Decomposition (SVD) to reduce dimensionality:[17]

Table 2.3: Second Order Co-occurrence Matrix, optimized with SVD

	child	dog	toast	cat	puddle
happy	325	128	0	0	0
big	0	145	0	153	0
sunny	0	0	76	23	0
mud	0	92	0	0	163
raisin	0	0	32	0	0

[17]fictional word counts and weights, representing the analysis results of a fictional corpus, are used for the entire 2nd order example

In Table 2.3 each entry is now a feature weight (not a word count) and represents each word's importance in the overall corpus.

- the words "happy" and "child" are first order co-occurrences: they appear as a bigram in the corpus
- the vectors for "happy" and "mud" are second order co-occurrences: "happy mud" does not appear as a bigram in the corpus, but their vectors exhibit an abstracted similarity (a second order similarity) because both words appear in a sentence with "dog"

Step 2: Using sent-1 as an example ("The happy child and the big dog found a mud puddle."), each word in sent-1 that has a corresponding matrix row in Table 2.3 is replaced by that row. This set of words (row) is the vector that represents that word's conceptual connections within the corpus. Three of the bigrams in sent-1 are still present in the matrix after SVD optimization:

Table 2.4: Second Order Co-occurrence: sentence one vectors

	child	dog	toast	cat	puddle
happy	325	128	0	0	0
big	0	145	0	153	0
mud	0	92	0	0	163

and:

"The HAPPY child and the BIG dog found a MUD puddle."

becomes:

The [**child, dog**] child and the [**dog, cat**] dog found a [**dog, puddle**] puddle.

Step 3: The average is taken across all 3 vectors to transform this second order co-occurrence into a second order representation. The resulting weights now represent the conceptual connections of this sentence within the overall corpus.

Table 2.5: Second Order Representation: sentence one vectors

	child	dog	toast	cat	puddle
sent-1	108.3	149	0	62.3	224

Frequency distribution (fdist) matrix:

This matrix is a term frequency representation of the data; each row is a document (sentence, in this project) and each column is a word. N-grams and n-gram window techniques can be used in conjunction with an fdist matrix to determine which terms are most strongly connnected to other terms in the corpus. N-gram models are used to create a picture of which words we can expect to see next to the current word: a bigram model for '*n = term_1*' would show us which terms appear as '*n+1 = term_2*', the term that directly follows *term_1*. Collocations and filters are NLP tools that can be used to find out how important each bigram is to every other bigram, and to the overall document as well [14].

Term frequency-inverse distribution function (tfidf):

This function is an alternative to the fdist described above. The matrix created represents a weighting of conceptual strength across documents. Each term is weighted for its importance in the overall corpus. A word that occurs often or rarely will likely not contribute to the overall semantic meaning of the text; therefore, rare, overly-prolific, and non-informative words will have a low rating. For example, if a word such as "Klingon" appeared once in a collection of 600 documents, each

containing approximately 3000 sentences, it would be removed in the tfidf analysis. Such an isolated, specialized word would be removed during later analysis anyway - there is simply nothing to connect this word to a useful number of other words - better to remove it before the more computationally expensive step of linguistic analysis. On the other hand, words such as "is" are usually prolific enough to be removed as well. Such words can appear so many times that they would fill the top levels of analysis and subvert more relevant concept connections.

Tokenization and noise reduction tools

The following NLP tools are described in more detail in Chapter 3:

1. Tokenization: break a stream of text symbols into words, phrases, symbols, or other meaningful elements; tokenize a document into sentence or tokenize sentences into words.

2. Stripping: remove punctuation and stopwords (common words such as "the" and "a").

3. Stemming: reduce each word, where possible, to its root form, and remove capitalization.

4. Indexing: after all of the above, including the removal of many sentences, each remaining sentence must be tagged with its original document and sentence number.

Software and model examples

Software and models in this section use supervised or semi-supervised machine learning. They also use any or all of: part-of-speech (POS) tags, labelled text, tagged

dictionaries.

The Stanford NLP Group focuses on probabilistic approaches that include both statistical and NLP models, utilizing both supervised and semi-supervised machine learning: part-of-speech (POS) tags, tagged text, and hand-tagged dictionaries. The group maintains many online resources, including a part-of-speech tagger and probabilistic parser.[18]

SenseClusters, as mentioned in Section 2.2.2, is a graduate project created by Professor Tom Pedersen and his graduate students at the University of Minnesota in 2002-2004. It is a semi-supervised machine learning model that utilizes a tagged dictionary (WordNet) for weights and disambiguation, as well as a tagged benchmark corpus for testing. SenseClusters is an LSI model that utilizes SVD[19] for dimensionality reduction and focuses on second order matrix and feature selection for concept clustering [16].

SCIgen is a Graduate research project created at MIT CSAIL[20] [20]. SCIgen generates random computer science research papers, which have been submitted to, and accepted by, conferences worldwide. The project uses unsupervised machine learning and a basic sentence structure to generate its research papers. The SCIgen template was modified and relaxed to create the linguistic template for the Automatic Generator.

PyProse: Formerly MacProse, PyProse [6] uses a dictionary that is completely unstructured. The purpose of PyProse is to generate poetic metre, not to generate cognizant output; therefore, the output produced was too nonsensical to use in this project (words are not even loosely grouped into categories):

> To rise couldn't repair the waste, and to emerge was the camp of paint

[18] http://nlp.stanford.edu/software/tagger.shtml
[19] Singular Value Decomposition
[20] https://pdos.csail.mit.edu/archive/scigen/

between the vortex and a spot.

Since any orchestra below a farmer would walk, why have they raced?

Although the stone has competed, how have so greater a driver talked?

The next two programs do not create original content, they use text that is culled from the web.

The Apostrophe Engine: 1994 Bill Kennedy created a poem titled "Apostrophe (1994)" which consisted of a list of sentence fragments culled from the web, each beginning with the bigram "you are". The poem became the homepage of the Apostrophe Engine. Originally, each fragment was a hyperlink that, when clicked, would prompt the site's search engine to scan the web for new sentence fragments beginning with the words "you are" and containing default keywords from the fragment. Each sentence fragment was terminated when the engine encountered a period.

Currently the apostrophe engine is offline. In the words of Bill Kennedy: "We speculated from the outset that once sections of the book began to appear online, the engine would begin to cannibalize itself, returning its own results before other, less likely matches."[21] [10]. Output from the Apostrophe Engine, while it was still online:

you are a soldier

you are implying- Sherlock Holmes: I'm not implying anything

you are an idiot

you are the one who shot him

you are choosing to ignore anything you see that doesn't comply with it

you are not serious

The Apostrophe Engine was not suitable as a text generator in this project as the

[21] http://www.apostropheengine.ca

output text was both culled from the web and had the added restriction of being delimited by "you are". These two words can be removed from each sentence but doing so leaves the majority of sentences unusable, as shown below:

> a soldier
> implying- Sherlock Holmes: I'm not implying anything
> an idiot
> the one who shot him
> choosing to ignore anything you see that doesn't comply with it
> not serious

Poetry Machine A text generator created by David Link that uses word association and semantic relationships. Poetry Machine can be initialized using its own random word generator or using words entered from a keyboard. The generator uses text culled from the web.[22]

2.5 Summary

- This project is based on three principles:
    1. pseudo-random text generation that preserves linguistic integrity at a basic new-language-learner level
    2. unsupervised machine learning for architecture portability and evolution
    3. low computational cost
- Statistical algorithms:

[22] http://alpha60.de/art/poetry_machine/

  - VSM, Vector Space Model: bag-of-words representation; document-term matrix, cosine similarity clustering
  - LSI, Latent Semantic Indexing: uses SVD and tfidf to reduce dimensionality and optimize concept visibility.
  - PLSI, Probabilistic Latent Semantic Indexing: a generative implementation of LSI that maximizes the joint probability of documents and terms in a corpus. PLSI estimates the probability distribution of each document on the hidden topics independently and the number of parameters in the model grows linearly with the size of the corpus. Problems with overfitting.
  - LDA, Latent Dirichlet Allocation: implements the EM algorithm of PLSI with a Dirichlet distribution. The probability distribution of each document, over topics, is treated as a hidden random variable. Overfitting is now under control.
  - PLSI and LDA both discover hidden topics, but only in Euclidean space. Recent studies suggest that the embedded sub-manifold is a more accurate representation of the document space [10][23].
  - LapPLSI: Laplacian Probabilistic Latent Semantic Indexing: implements a regularizer variable and other custom parameters in the EM algorithm of PLSI. LapPLSI models the document space as a submanifold embedded in the ambient space, revealing hidden semantics and deeper concept connections beyond Euclidean space.

- NLP tools:
  - basic document processing tools: tokenizing, stopword / punctuation removal, word stemming, capitalization normalization, matrix indexing

  - word frequency and placement analysis: frequency distribution (fdist) matrix (term frequency representation), n-grams, collocations
  - term frequency-inverse distribution function (tfidf) matrix: each entry is now a feature weight (not a word count) that represents the importance of each word/term across the entire corpus

- No part-of-speech (POS) tagging or established relationships between words. This project uses a modified SCIgen document template and a custom dictionary (word list only). To begin completely from scratch, with no word grouping, while intriguing, is a much larger project.

- Document clustering (for this project sentence clustering) on the embedded submanifold. The hidden topics extracted by the topic modelling approaches can be regarded as clusters, and the conditional probability density of each cluster $P(z_k|d_i)$ can be used as the cluster label (topic) of each document (sentence).

- Dimensionality reduction: NLP tools process the document to reduce noise (punctuation, stopwords, etc) and either a tfidf or an fdist matrix is created.

- Stable evolution of learning LapPLSI uses distribution-based clustering, reducing the computational cost of time and space. This allows a more frequent aggregation and re-processing of ALL current data, avoiding the instability associated with integration of new weights and similarity results with old.

Chapter 3

Architecture of the Automatic Generator

The objective of this project is to design an unsupervised machine learning architecture that can gather meaningful concepts, at low computational cost, from randomly generated text and link these concepts together in a poem. To do this, an architecture was designed consisting of five discrete components (blocks) and a set of custom tools, parameters, and metrics. These are combined with proven methods of both statistical inference and NLP (linguistic) analysis. The metric for success is the automatic generation of poetry, that can pass as human-generated, as assessed by 20 volunteers.

A varying number of documents are generated, concatenated, and pre-processed in Block 1, to create the matrix and sentence files for input to Block 2. In the experiments documented in this chapter, concatenated documents of 100, 300, 600, and 1200 original documents were created from each dataset, resulting, on average, in a matrix containing:

Table 3.1: Document-Term Matrix, average stats

	size (kb)	rows (sentences)	columns (words)	cells
100 documents	4	2 300	1 800	1 400 000
300 documents	14	2 900	4 800	4 600 000
600 documents	30	3 400	8 700	10 000 000
1200 documents	71	4 400	15 900	23 500 000

In Block 3 each of these document sets was then run through a set of parameter options: 5, 7, and 10 latent topics/clusters, and multiple probability thresholds, producing optimized, reduced files. In Block 4, an fdist matrix was created for each reduced file and trials were run for bigram and trigram options, each with Pointwise Mutual Information (PMI) and Logistic Regression (LR) analysis, as outlined in Section 2.4. The following table shows the average number of words and sentences in the Block 4 fdist matrix, 5 cluster, 600 document trial. The average number of unique words falls within the optimum range (100 unique terms) for the NLP analysis performed in this block [11]:

Table 3.2: Document-Term Matrix: 600 documents, average stats

	size (kb)	rows (sentences)	columns (words)	cells	unique words
5 clusters	288	424	647	287 809	200

The original data is generated using pseudo-random methods: pseudo in the sense that the text produced is neither grammatically correct, nor is it completely nonsensical. The text reflects a sociolinguistic balance: a basic level of language structure and word placement as would be normally assimilated by young children, or by travellers in a foreign country. An augmented SCIgen template [20] and

customized dictionary (terms only, no tags) was used. Basic patterns emerge from this structure that can be built on:

> **Block 1**: Generate dictionary. Generate text.
> Create and compile the dictionary. Create the data using the modified SciGen template and the custom dictionary.
>
> **Block 2**: File processing. Noise reduction. Create viable sentences.
> Process files using basic NLP analysis techniques such as stripping, stemming, and stopword removal. Create files of viable sentences for input to Block 3.
>
> **Block 3**: Statistical Analysis.
> Reduce noise even further by creating term frequency-inverse distribution function (tfidf) matrix. Analyze the files with the embedded submanifold techniques used in Deng Cai's LapPLSI algorithm [2]. Adjust sentence choices with threshold probability parameters.
>
> **Block 4**: NLP Analysis.
> Analyze file with advanced NLP analysis techniques such as n-grams, collocation filters, pointwise mutual information (PMI) and logistic regression (LR).
>
> **Block 5**: Evolve concept visibility parameters. Polish the poems.
> Implement patches to polish the poems. Evolve the sociolinguistic concept connections.

In this project we have implemented Blocks 1-4. With experiment we found that the best results were produced by using files of 600 concatenated documents, clustered over 5 hidden topics and analyzed with LapPLSI, bigrams and mutual information techniques. These files produced the final matrix of viable sentences for creation of the poems. We can expect even better results with larger raw input files.

3.1 The Challenges

Linguistic integrity: To retain the semantic structure of the data requires the retention of not only the words in a sentence, but also the underlying meaning of the sentence itself. Optimal retention of semantic structure (linguistic meaning) must be balanced with the computational costs of space and time.

Dimensionality reduction: The computational cost of space and time necessary for dimensionality reduction must be balanced with algorithmic options for optimal retention of semantic and syntactic integrity (optimal concept retrieval). Techniques such stemming, stopword removal, and tfidf matrix creation help to reduce dimensionality with minimal impact on text semantics [19][13].

Stable evolution of learning: Both dimensionality reduction and linguistic integrity contribute to a stable evolution of learning, defined in this context as the accuracy of analysis results over time.

3.2 The Approach

To address semantic integrity we consider the order of stochastic analysis, and the linguistic structure of the input data. For the order of stochastic analysis the first question is how to approach dimensionality reduction. Either statistical or semantic (NLP) analysis must be used for this important step in processing.

Using statistical inference as a first step, as we have done in this project, creates a tfidf (term frequency-inverse distribution function) matrix and a Probability Distribution P_D, reduces noise, and increases concept visibility. Statistical inference also improves disambiguation, finding relationships between sets of information, or concepts (co-occurrences, for example), that are often eliminated when semantic methods are used as a first step [19]. NLP tools can then be used to find the ways

in which these concepts are inter-related. Take for example the following text:

"blind venetian" and "venetian blind"

Lexical analysis would not connect the two sets of words but statistical analysis preserves them in the data set. Statistically independent data is always linguistically unrelated BUT linguistically unrelated data is not always statistically independent [15].

To preserve linguistic integrity we assess various facets of language structure and word placement. Maximizing retention and visibility of the text's inherent underlying meaning must be balanced against the over-use of language constructs. The goal is to strike a sociolinguistic balance between nonsensical text and rigid grammatical structure. For example: "dog big" may be a first attempt by a young child but exposure to native speakers would organically evolve the description to "big dog".

To address dimensionality reduction we look at the computational time/space requirements and optimal ways of reducing to useful data. Document representation with unsupervised machine learning produces a naturally sparse matrix of high dimensionality (one dimension for each term in the corpus), making dimensionality reduction an important consideration. Statistical analysis as a first step is an effective optimization method, reducing data volume, and increasing the strength of conceptual relationships.

Raw versus processed data is also a consideration. Raw data includes punctuation, stopwords, newline characters, all versions of words, duplicate sentences, etc. In our trials the sheer volume of data in each raw text file created an unmanageably large matrix when run on the available resources, severely limiting the size of the data samples that could be used, and making relevant comparisons between raw and processed data unrealistic. There is some debate in the document analysis

community about whether stemming, stopword removal, and punctuation stripping improves output [15], but all three methods considerably reduce data noise, allowing larger datasets to be processed.

There is also no evidence in the literature to conclude that running raw data through the architecture, even if the computing resources were available, would produce better results. The process of dimensionality reduction naturally removes the majority of stopwords and punctuation from raw data, even if this information is not removed in pre-processing, but it also leave behind some connector words, such as "at" or "therefore", in the raw data. These words incur a higher analysis cost at later stages. Adjusting n-gram windows in raw text can help, as is shown in the following example:

Starting with the raw text sentence:

The dog has a toy.

We first run a dimensionality reduction algorithm to find the co-occurrences of "dog" and "toy" in the sentence. A trigram analysis with an n-gram window[1] is required. Processing a file and using trigram windows is computationally expensive: A terms produces A^3 trigrams plus the n-window permutations. Processed text, as opposed to raw text, requires only bigram analysis (A terms produce A^2 bigrams) to catch "dog toy", "dog has toy", "dog and toy", etc. Running trigrams on processed text did not produce better results, indicating that the Block 2 processing protocol creates bigram optimized data.

To address stable evolution of learning we look at clustering options that maximize relevance, speed, and the stability of results. To create a stable evolution of learning, all data is processed on every run: new data is aggregated with old and

[1] analyzing groups of words, looking for 2 keywords separated by a non-keyword window of n words

fresh cluster results are produced, allowing analysis to start fresh every time the algorithm is run, with no pre-conceived idea of cluster topics and cluster members from past trials. If new data is always run discretely, and the results integrated with previous results, the overall outcome can become skewed [18]. The low cost of scaling the LapPLSI-optimized algorithm means data can be aggregated and re-clustered frequently.

3.3 The Architecture

The architecture consists of 5 blocks, each a discrete part of the Automatic Generator. In this section we outline, for each block:

- What the Block does
- Why it is important
- How it fits into the big picture

Figure 3.1: Block 1: Create dictionary. Generate text

The purpose of Block 1 is to generate carefully structured pseudo-random text: create and compile the dictionary, generate files of raw, pseudo-random text using the modified SciGen template. This block regulates linguistic integrity and language level.

procedure RUN DICTIONARY PROGRAM
create custom dictionary.
overwrite SciGen dictionary with custom dictionary.
modify and customize SciGen template.
end procedure

procedure RUN MODIFIED SCIGEN PROGRAM
use dictionary and template to generate files of raw, pseudo-random text.
generate number of documents required.Write each to file. Concatenate.
end procedure

Figure 3.2: Block 2: Create Viable Sentences

The purpose of Block 2 is to create a file of usable sentences from the pseudo-random text generated in Block 1. Custom NLP tools and techniques are used to reformat, clean, delimit, tokenize, and index each file. Matrix sparsity (noise) is also reduced through these techniques.

```
procedure PROCESS FILE
    for each sentence in file do
        tokenize to sentences
        tokenize to words
        strip punctuation
    end for
end procedure

procedure LOCATE VIABLE SENTENCES
    for each sentence in file do
        keep viable sentences; write to file          ▷ 3-12 words
    end for
end procedure

procedure REMOVE SYNTACTIC MARKERS
    for each sentence in file do
        normalize
        strip stopwords                          ▷ augmented algorithm
        stem                                     ▷ augmented algorithm
        index                                    ▷ index words to sentences
    end for
end procedure
```

Figure 3.3: Block 3: Inference Analysis

Use statistical analysis tools to identify and extrapolate concepts: The purpose of Block 3 is to identify deeper structural patterns in the text through statistical analysis techniques. The output of Block 2 is analyzed using the LapPLSI algorithm for embedded sub-manifold analysis [2] outlined in section 2.2.4.

procedure ANALYZE WITH LAPPLSI
    for each file **do**
        create tfidf ▷ weighting: how important is each sentence
        extract joint probabilities ▷ run LapPLSI
    end for
end procedure

procedure CLUSTER AT PROBABILITY THRESHOLD
    retain sentences above designated Probability Threshold
end procedure

Figure 3.4: Block 4: Natural Language Processing (NLP) Analysis

Use semantic tools to identify and extrapolate concepts: The data from Block 3 is prepared for analysis (similar to the processing in Block 2), then analyzed with NLP tools to extract groups of conceptually-connected sentences for the poems.

```
procedure PROCESS FILE
    for each sentence in file do
        tokenize to sentences
        tokenize to words
        strip punctuation
    end for
end procedure

procedure REMOVE SYNTACTIC MARKERS
    for each sentence in file do
        normalize
        strip stopwords                          ▷ augmented
        stem                                     ▷ augmented
        index                                    ▷ index words to sentences
    end for
end procedure

procedure ANALYZE THE FILE SEMANTICALLY WITH NLP TOOLS(cluster sentences with related semantic concepts)
    for each file of sentences do
        create fdist
        for each option: bigrams, trigrams do
            Pointwise Mutual Information
            Logistic Regression
        end for
    end for
end procedure
```

Figure 3.5: Block 5: Fine-tune sentence creation and selection.

The purpose of Block 5 is to assemble the poem (lines, stanzas, author) and to evolve the HCI cognition-response by auto-generating fitness values for specific fine-tuning objectives.

```
procedure Auto-generate fitness values
    for each socio-linguistic goal do
        define contributing parameters
        combine in fitness formula
    end for
end procedure
procedure Analyze the output
    for each file do
        edit repetition
        run fitness formula
        cluster pre-determined number of conceptually related sentences
    end for
end procedure
procedure Assemble and polish poem(stanzas, formatting, author)
    for each file do
        cluster stanzas based on pre-determined parameters
        add blank line between stanzas
        add punctuation, capitalization
        generate author
    end for
end procedure
run analytics
```

Chapter 4

Implementation Detail

4.1 Block 1: Create the dictionary and the data

The purpose of Block 1 is to generate carefully structured pseudo-random text. A modified SCIgen template and a custom dictionary are used to generate the text files.

The project is based on unsupervised machine learning: no hand-tagged training sets, no pre-tagged data. This means that the Stanford Part-of-Speech-Tagger[1], WordNet[2], Visual Thesaurus[3], and similar online tools were not suited to this project.

For the dictionary this meant not using WordNet or similar tagged dictionary databases. The dictionary for this project is a wordlist only combination of three dictionaries: my own custom, and the dictionaries from the SciGen and PyProse programs. This block regulates the linguistic integrity (language level), semantic integrity (basic, cultural, and social vocabulary), and the syntactic integrity

[1]https://nlp.stanford.edu/software/tagger.shtml
[2]https://wordnet.princeton.edu/
[3]https://www.visualthesaurus.com/

(language-learner word order). For example: “The red car raced down the road.” is fine; “Raced road car red the.” is not.

The input text itself needed to be pseudo-random but not nonsensical; basic word order and social / cultural entries, in the spirit of a new-language-learner syntax, as documented in Section 3.3. The basis for this structure can be found in [3].

4.2 Block 2: Create viable sentences

The purpose of Block 2 is to create a file of viable sentences from the text generated in Block 1.

Custom NLP tools and techniques are used to reformat, clean, delimit, and concatenate each file.

- Remove syntactic markers: basic processing and reformatting of data to cluster-ready: tokenize to sentences, tokenize to words, normalize to lower-case, strip punctuation and stopwords, stem to root words.
- Create an index - need to link the words back to their original sentence(s).
- Cluster viable sentences. Write to file: minimum 3 words, maximum 12 words: sentences of 1-2 words increase data noise and rarely comprise a viable sentence (“I am” is one of very few) and more than 12 words is very long for a line of poetry and did not increase the quality of viable sentences in our results.

Note: stripping punctuation and stopwords, as well as stemming to root words, decreases matrix size at the cost of increasing ambiguity (for example, ‘stocking’ and ‘stocks’ both stem to ‘stock’). The statistical analysis of Block 2 and Block 3 restores disambiguity.

TOKENIZE to sentences:
[‘Experts regularly measure the place of agents.’, ...]

TOKENIZE to words:
[[‘ ‘Experts’, ‘regularly’, ‘measure’, ‘the’, ‘place’, ‘of’, ‘agents’, ‘.’ ’] ...]

STRIP punctuation and CULL sentences to desired min/max:
[[‘ ‘Experts’, ‘regularly’, ‘measure’, ‘the’, ‘place’, ‘of’, ‘agents”] ...]

NORMALIZE file (lower-case):
[[‘experts’, ‘regularly’, ‘measure’, ‘the’, ‘place’, ‘of’, ‘agents’]]

STRIP stopwords:
[[‘experts’, ‘regularly’, ‘measure’, ‘place’, ‘agents’]]

STEM with Porters Stemming Algorithm:
[[‘expert’, ‘regularli’, ‘measur’, ‘place’, ‘agent’]]

INDEX:
[[(3, ‘expert’), (3, ‘regularli’), (3, ‘measur’), (3, ‘agent’), (3, ‘place’)]]

Locating viable sentences of suitable length was also a challenge. Online tools like the Stanford POS Tagger provide grammatical sentence structures such as phrasal chunking (dividing sentences into phrases) and clausal grouping (groups of phrases) but a phrase is, by definition, a conceptual unit - one component of a clause - and therefore requires part-of-speech tagging in order to be identified.

4.3 Block 3: Distribution-based Analysis

The purpose of Block 3 is document representation and topic modelling through statistical inference. This Block uses the protocol outlined by [2] Deng Cai et al to analyze each file of documents (sentences) and cluster the related concepts using the LapPLSI protocol. Functionality of this block includes:

- Create a tfidf matrix: term frequency-inverse distribution function matrix represents a weighting of conceptual strength across documents. Each term is weighted for its importance in the overall corpus.
- Model the document space as a submanifold embedded in the ambient space.
- Reduce matrix dimensionality.
- Identify the topic models (related concepts) in non-Euclidean space
- analyze file of documents (sentences) and cluster the related concepts using the LapPLSI protocol. LapPLSI parameters used in this project:
  - number of documents in each trial: 100, 300, 600, 1200
  - number of topics / clusters in each trial: 5, 7, 10
- Create a probability distribution of the concept connections.
- Cluster sentences with related concepts (topics)
- Probability bar: Within each cluster (topic) keep the sentences that are most closely related. Each sentence that is kept will have a probability of connection to that cluster that is greater than, or equal to, a pre-set threshold. This step determines the sentences that are used in poem creation (Block 4).

Algorithm 4.1 Generalized EM Algorithm for LapPLSI

Variables: **N** = number of documents, **K** = number of latent topics

M = number of terms (words) in the dictionary

Number of nearest neighbours = p, Regularization parameter = λ

Newton step parameter = γ, Termination condition value = θ

Output: $P(z_k|d_i)$, $P(w_j|z_k)$, $i = 1, \ldots, N; j = 1, \ldots, M; k = 1, \ldots, K$

Compute the graph matrix W ([2] p 3);

Initialize probability distributions (parameters) $\psi_0 = P(z_k|d_i)_0, P(w_j|z_k)_0$;

$n \leftarrow 0$;

While (true)

E-step: Compute the posterior probability ([2], p3);

M-step:

Compute $P(w_j|z_k)_{n+1}$ (re-estimation equation ([2], p3);

Compute $P(z_k|d_i)_{n+1}$ (re-estimation equation ([2], p3);

$P(z_k|d_i)^{(1)}_{n+1} \leftarrow P(z_k|d_i)_{n+1}$;

Compute $P(z_k|d_i)^{(2)}_{n+1}$ (Geometric Regularization Equation 2.5);

While$((\varphi(\Psi^{(2)}_{n+1}) \geq \varphi(\Psi^{(1)}_{n+1}))$

$P(z_k|d_i)^{(1)}_{n+1} \leftarrow P(z_k|d_i)^{(2)}_{n+1}$

Compute $P(z_k|d_i)^{(1)}_{n+1}$ (Geometric Regularization Equation 2.5);

If $(\varphi(\Psi^{(1)}_{n+1}) \geq \varphi(\Psi_n))$

$P(z_k|d_i)_{n+1} \leftarrow P(z_k|d_i)^{(1)}_{n+1}$;

Else

$\Psi_{n+1} \leftarrow \Psi_n$;

If $(\varphi(\Psi_{n+1}) - \varphi(\Psi_n \geq \theta))$; break;

$n \leftarrow n + 1$;

Return Ψ_{n+1}

4.4 Block 4: Natural Language Processing (NLP) Analysis

This block identifies patterns of repetition in the text by analyzing factors such as how many times a specific word appears in a specific document, and identifying co-occurrences of every corpus entry (word) with every other corpus entry, across all documents.

The purpose of Block 4 is to identify pattern recurrence within patterns. A tfidf matrix is created to normalize the data and create weights that reflect the overall (global) impact of each word combination. The data from Block 3 is prepared for statistical analysis (similar to the processing in Block 2), then analyzed with NLP tools to extract groups of conceptually-connected sentences for the poems. Patterns of bi- and tri- gram recurrence / placement are analyzed using collocation filters, Pointwise Mutual Information (PMI) and Logistic Regression (LR).

- As in Block 3, remove syntactic markers: basic processing and reformatting of data to analysis ready.
- Create a Frequency Distribution matrix (fdist) for each document.
- Create topics (clusters) of conceptually related bigrams and trigrams, using collocation filters, Pointwise Mutual Information (PMI) and Logistic Regression (LR) analysis.
- Re-index words to sentences.

4.5 Block 5: Evolve the HCI Cognitive Response

The purpose of Block 5 is to assemble the poem (lines, stanzas, author) and to evolve the HCI cognition-response by auto-generating fitness values for specific fine-tuning objectives.

Fine-tune the HCI response for sentence creation and selection:

- Smooth: optimize the sociolinguistic connections with patches.
- Improve and fine-tune sociolinguistic parameters.
- Format: author, line length, stanza length, etc.
- Define advanced sociolinguistic goals.
- Auto-generate fitness functions for specific fine-tuning objectives.

Chapter 5

Experimental results and analysis

This chapter presents a comparison study using various selection parameters, as set out in the Chapter 3 description of the Automatic Generator model. Specifically, we conduct a series of experiments to compare the strength of the concept connections produced when varying the size of the input dataset, the number of hidden topics (clusters), and the choice of natural language analysis tools.

5.1 Environment

Traditionally, such results are evaluated based on their similarity to benchmark and/or hand-tagged datasets. In the case of the embedded manifold code used here [2] Deng Cai et al cluster two well-known datasets, Reuters and TDT2 (Section 2.2.4), comparing their resulting topic labels against the hand-tagged benchmark labels of each dataset. Reuters and TDT2 are known as single label datasets: each document is hand-tagged with the one topic to which it most strongly belongs.

This project uses unsupervised machine learning and therefore has no labels to compare to a benchmark dataset. For our purposes, we evaluate the resulting

sentence files through reader feedback.

Probability-label versus single-label datasets

The goal of our model is to cluster conceptually similar sentences from dynamically created, pseudo-random, text. We begin with the understanding that hidden conceptual patterns will vary, within a dataset, in relation to variance in parameters such as number of topics, size of input file, the subjective cluster threshold, and the analysis tools used.

Therefore, in this project, we do not aim to improve the recognition of a benchmark label (news, entertainment, etc.) by increasing the size of the corpus for each label. Instead we aim to improve the accuracy of concept discovery by allowing multiple labels to be attached to each sentence, and allowing the cluster probabilities (the strength of the concept connections between these labels) to be re-computed at each instance of new data. The final clustering represents not the ONLY label to which the sentence belongs but the label to which it most strongly belongs.

The use of a probability distribution, rather than a single pre-tagged label, allows a range of clustering output goals to be addressed by both coarse- and fine-grained parameter adjustment. These goals include: threshold probability, n-grams, statistical analysis, secondary algorithmic reduction. Another advantage is low computational cost: the use of the embedded submanifold model allows the hidden topics to be found, and their similarity within the probability distribution to be utilized as an inherent clustering mechanism.

Analysis of Stability

If the AutoGen analysis is to remain relevant to a specific user, over time(evolve), the analytic results must evolve to reflect changes in user interest, knowledge, and understanding. This is a challenge for all NLP applications, as well as every applications that uses classifying/clustering. Results become unstable (skewed, unreliable) if historical probability values are simply integrated with new values. There are complex ways to increase stability but such methods also increase complexity. As mentioned above, LapPLSI's embedded sub-manifold model has a considerably lower computational cost than traditional stand-alone clustering algorithms, such as K-means, allowing for increased stability without increased complexity.

One of the main advantages of this functionality is an increase in the stability threshold. As the amount of aggregate data in each dataset varies, so does the depth of 'understanding', or the discovery of latent concepts. For example, a student could use their previous year's essays as the initial AutoGen input dataset. As the current year progresses, the student could quickly and easily recompile the AutoGen input dataset with new essays included. In this way the integrity of the Auto Gen results are preserved, also creating a stable evolution of learning.

5.2 Results of LapPLSI analysis in Block 3

Results of Block 3: clustering results after LapPLSI analysis and threshold culling.

5 clusters

Figure 5.1: 5 clusters, 100 Aggregated files

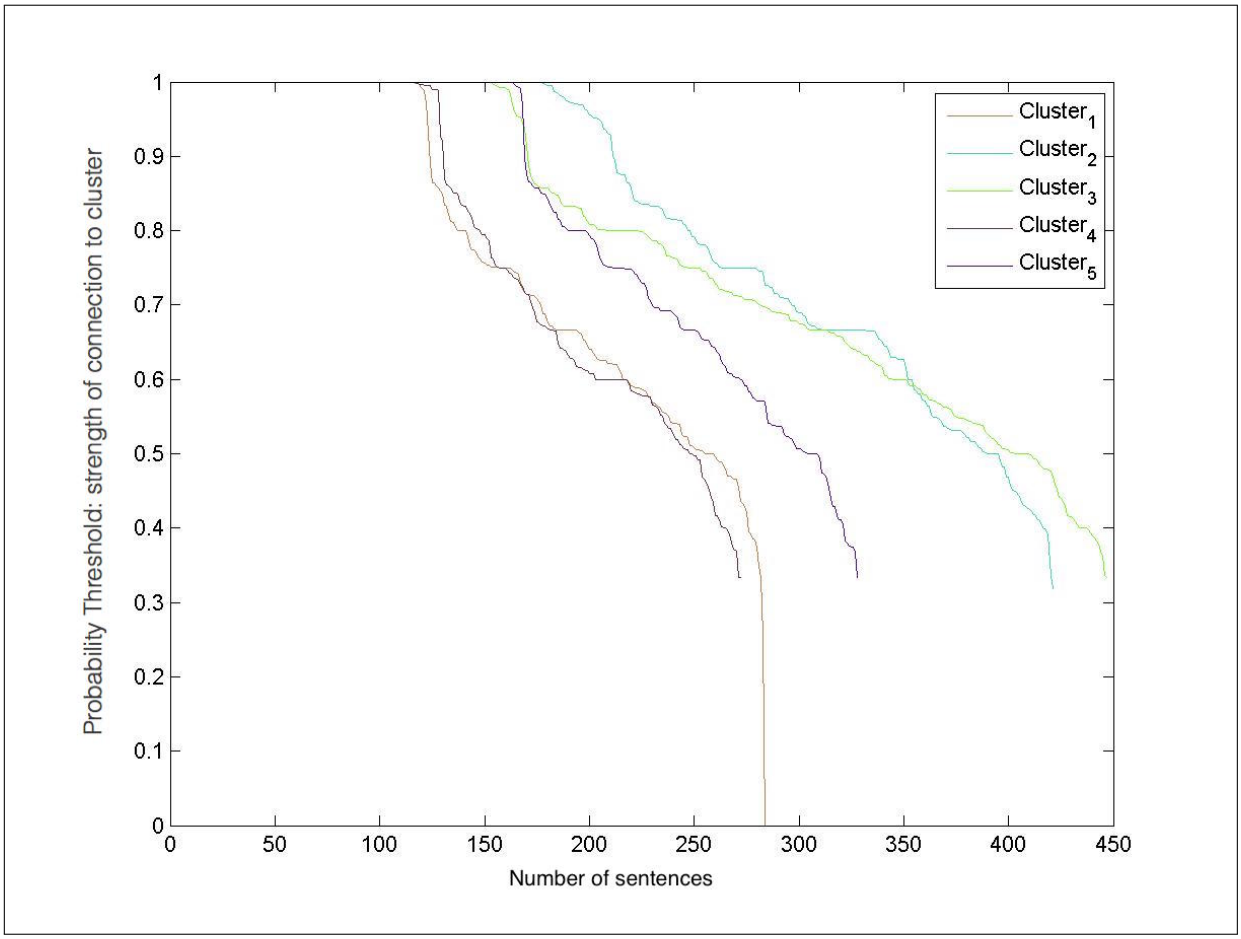


Figure 5.2: 5 clusters, 300 Aggregated files

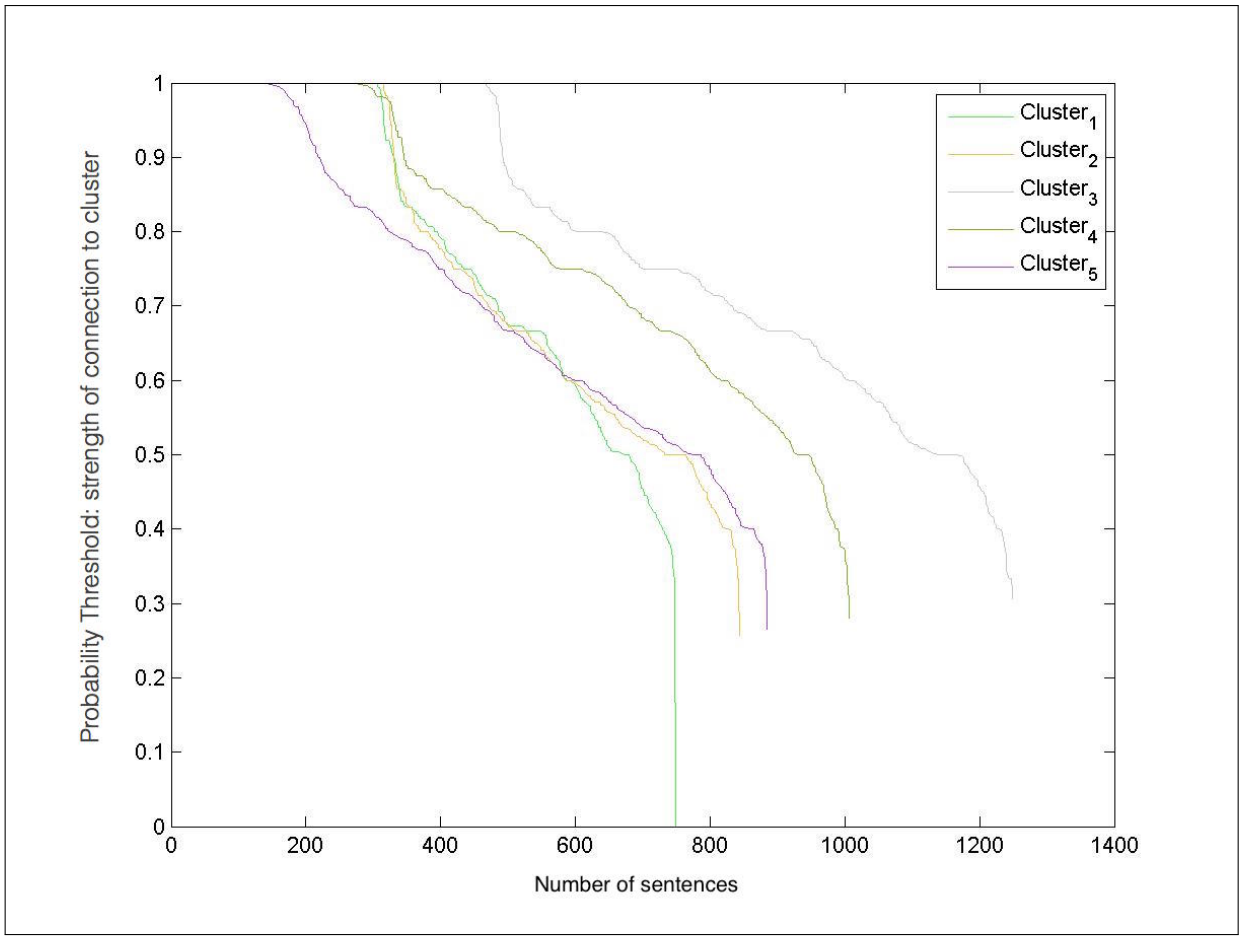


Figure 5.3: 5 clusters, 600 Aggregated files

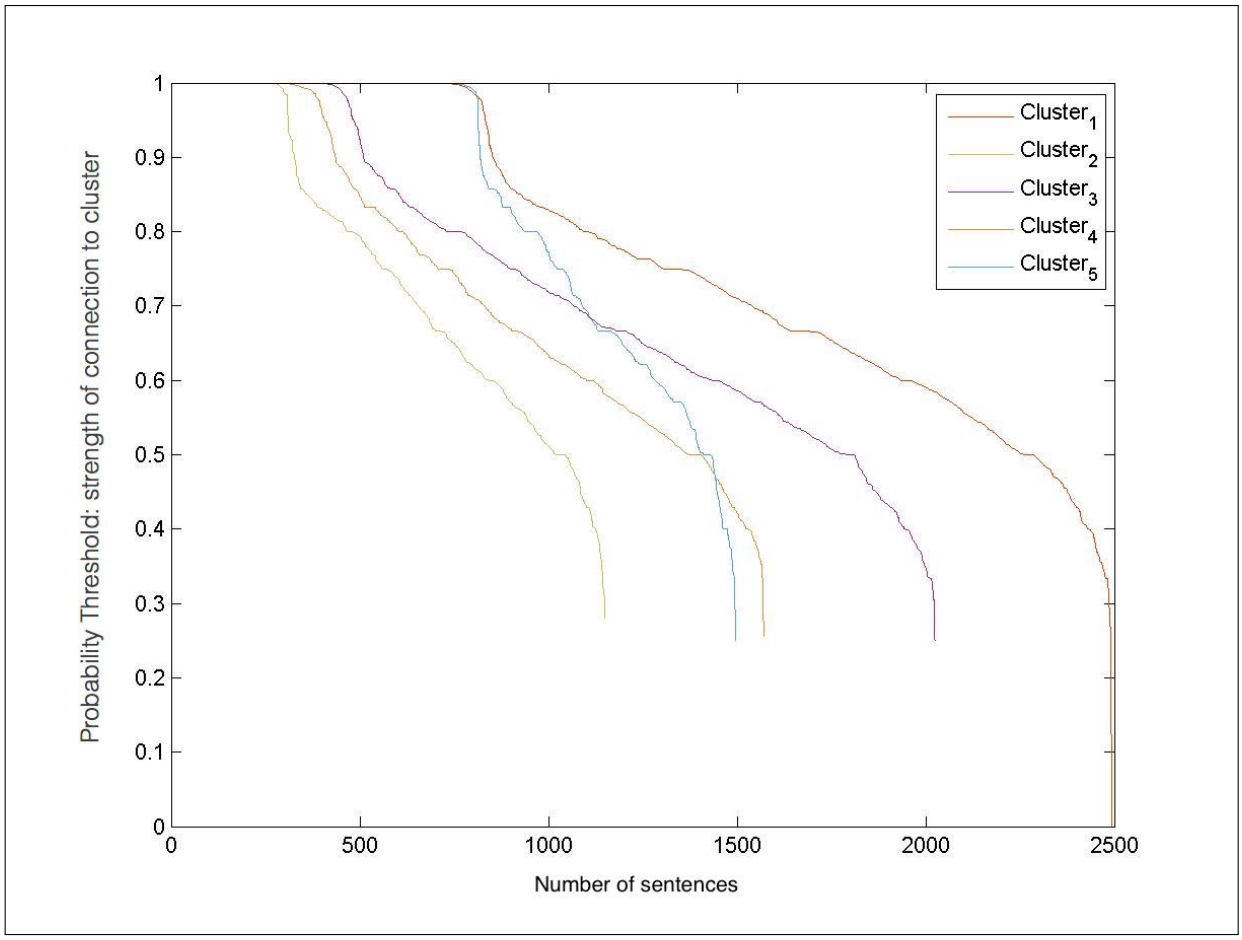
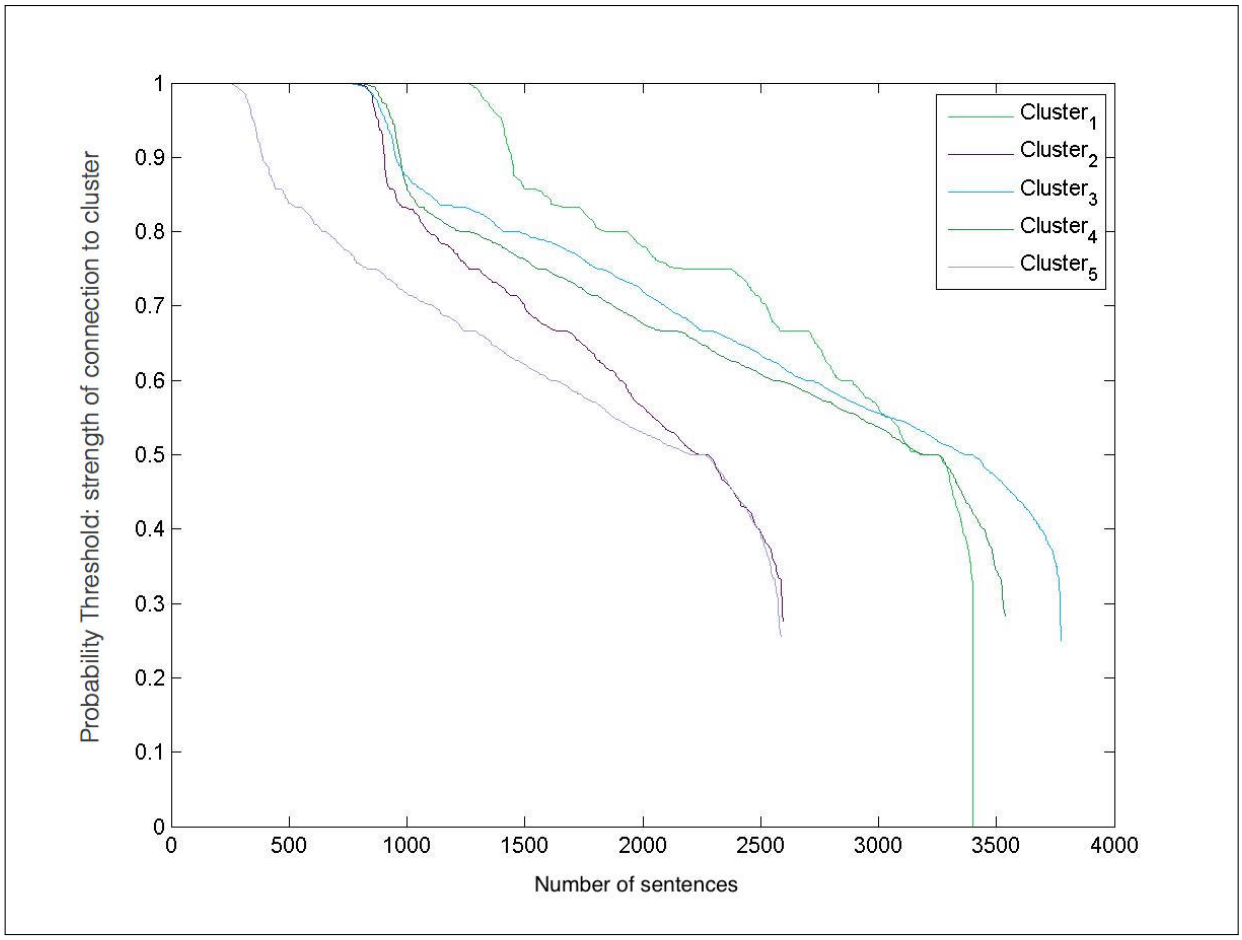


Figure 5.4: 5 clusters, 1200 Aggregated files



7 clusters

Figure 5.5: 7 clusters, 100 Aggregated files

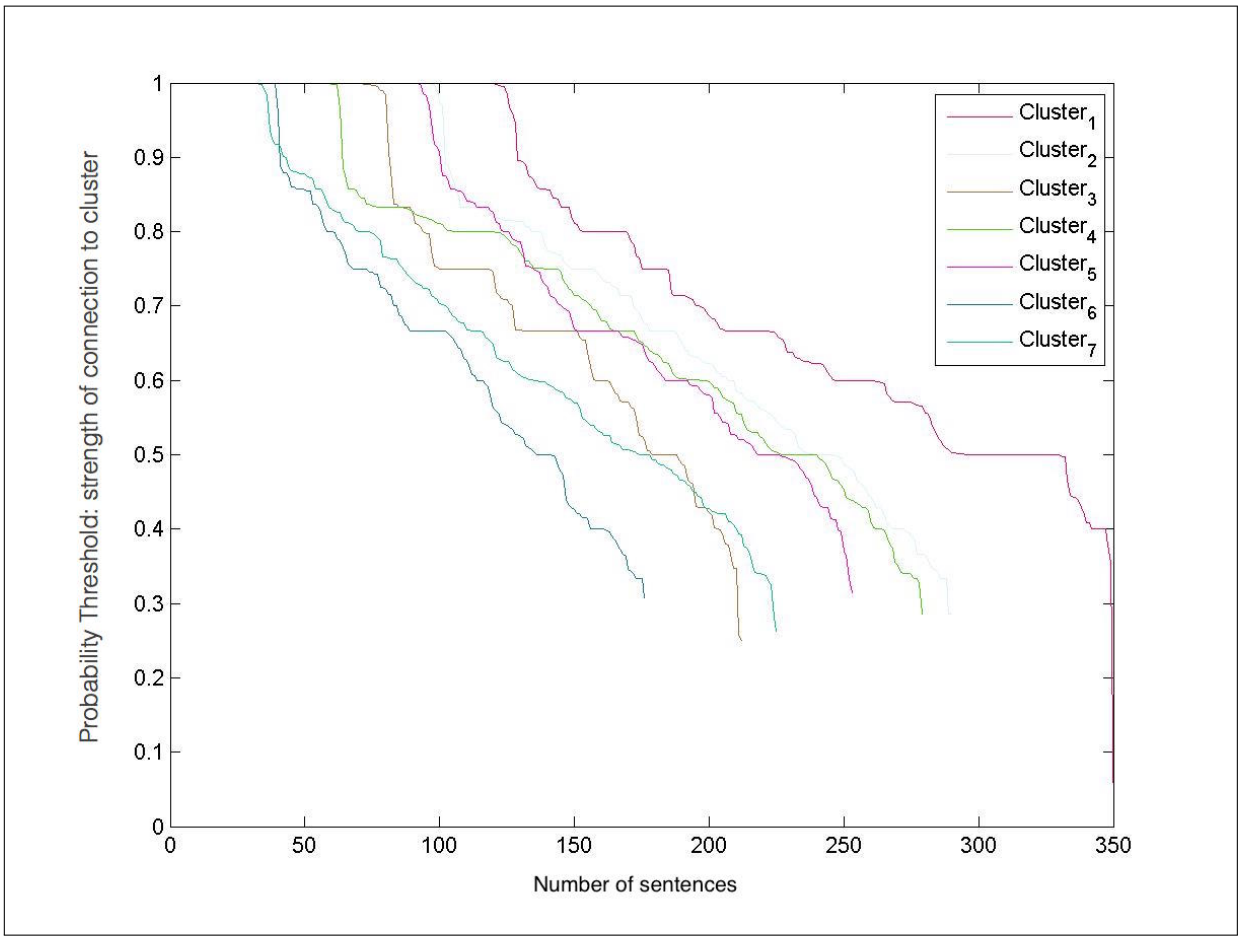


Figure 5.6: 7 clusters, 300 Aggregated files

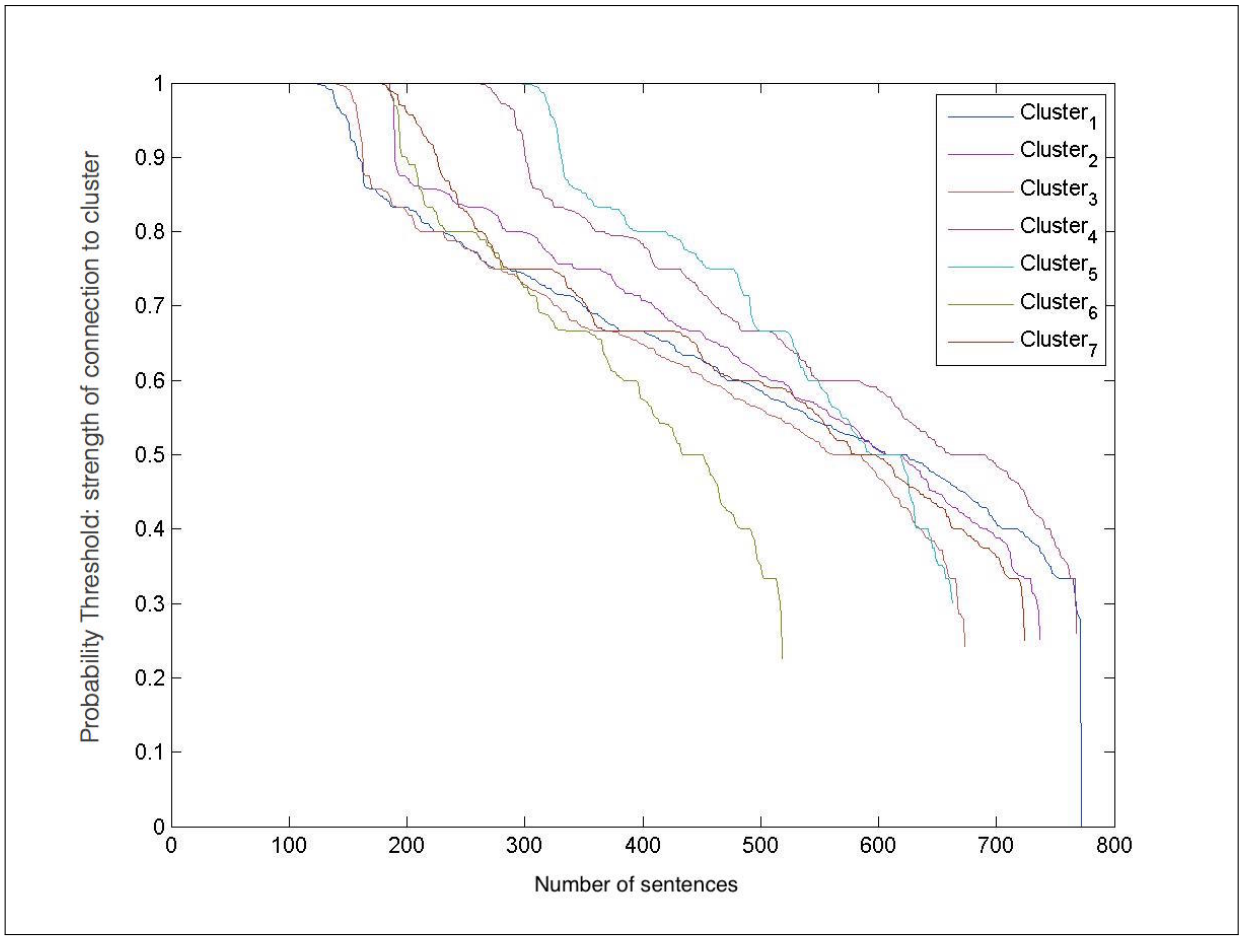


Figure 5.7: 7 clusters, 600 Aggregated files

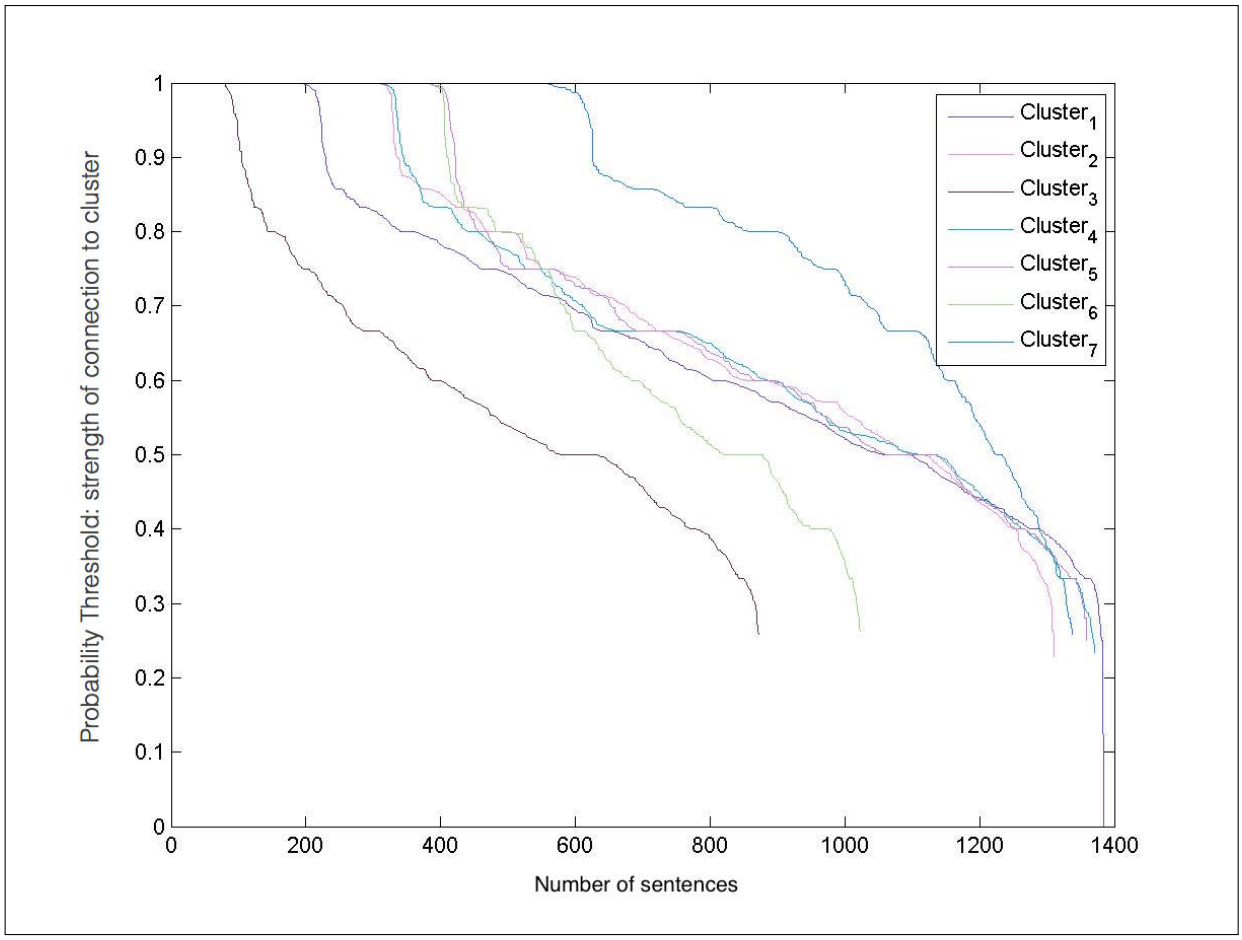
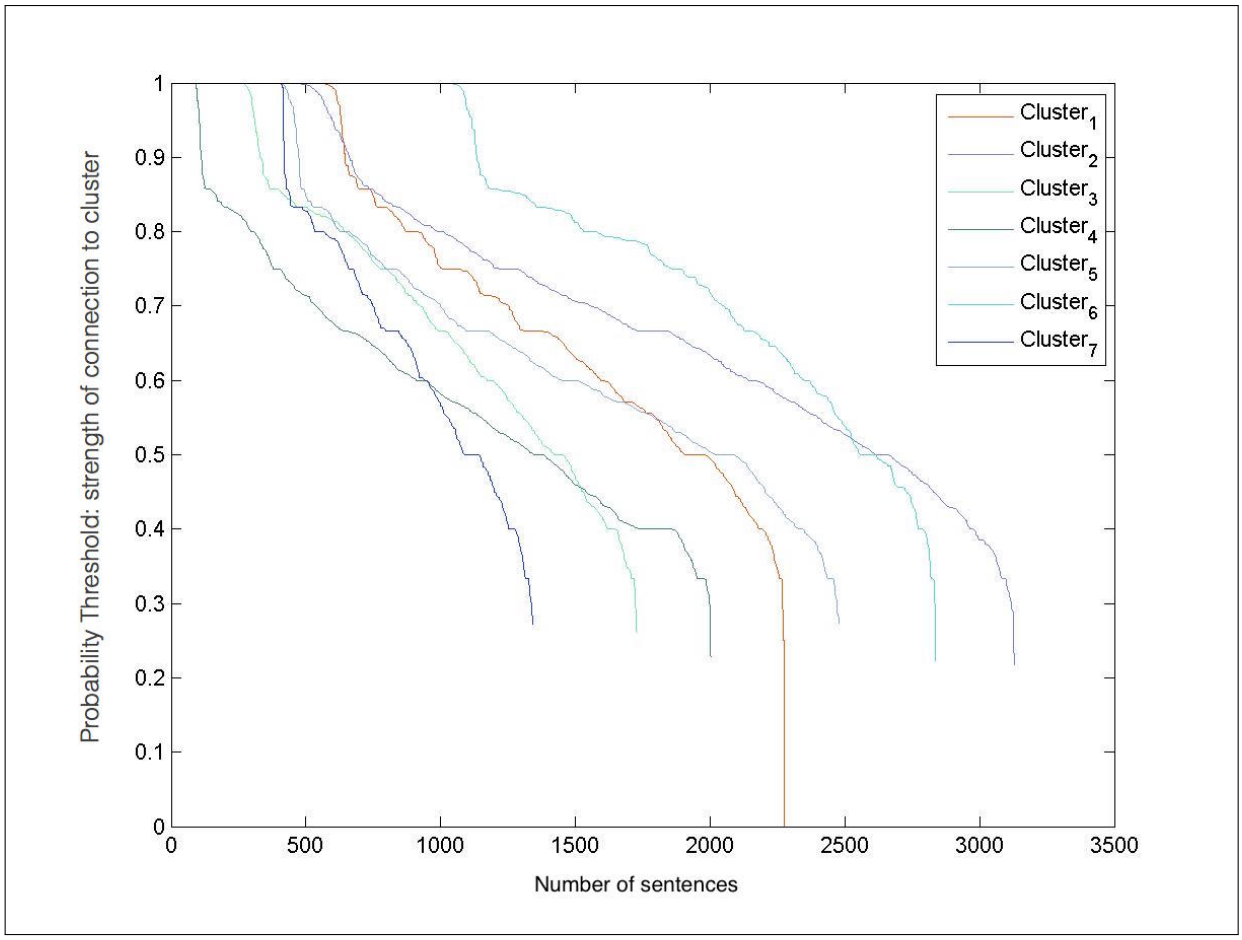


Figure 5.8: 7 clusters, 1200 Aggregated files



10 clusters

Figure 5.9: 10 clusters, 100 Aggregated files

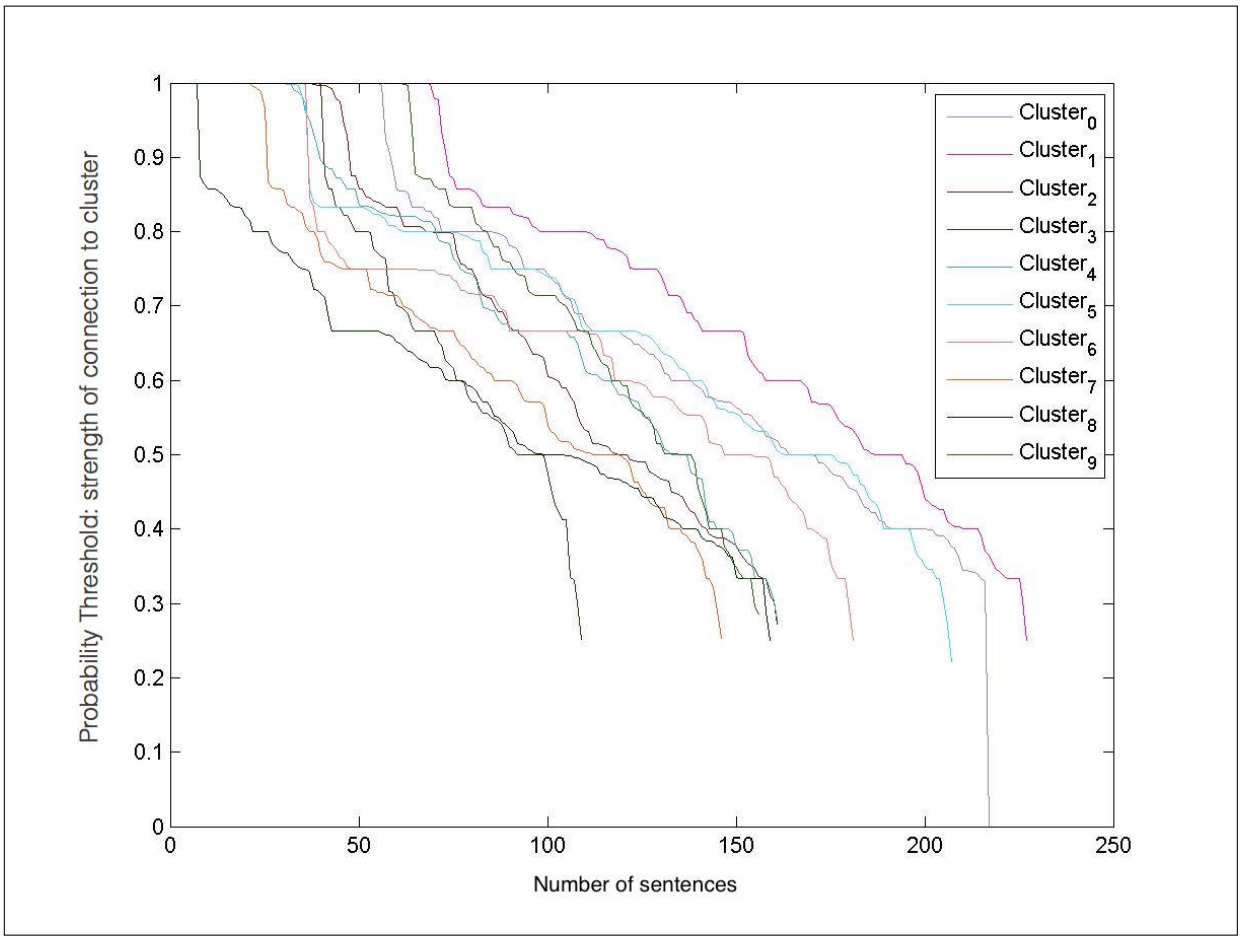


Figure 5.10: 10 clusters, 300 Aggregated files

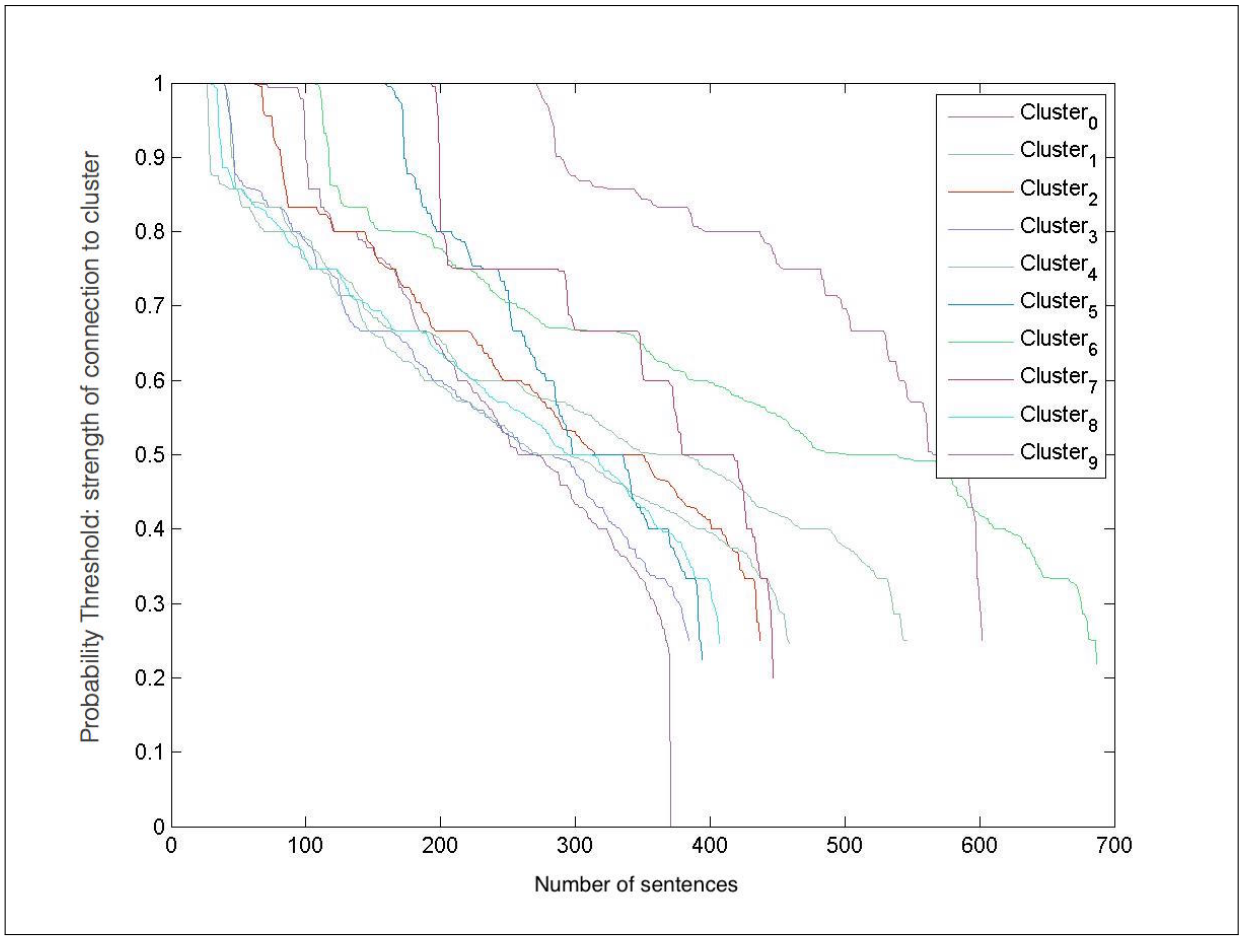


Figure 5.11: 10 clusters, 600 Aggregated files

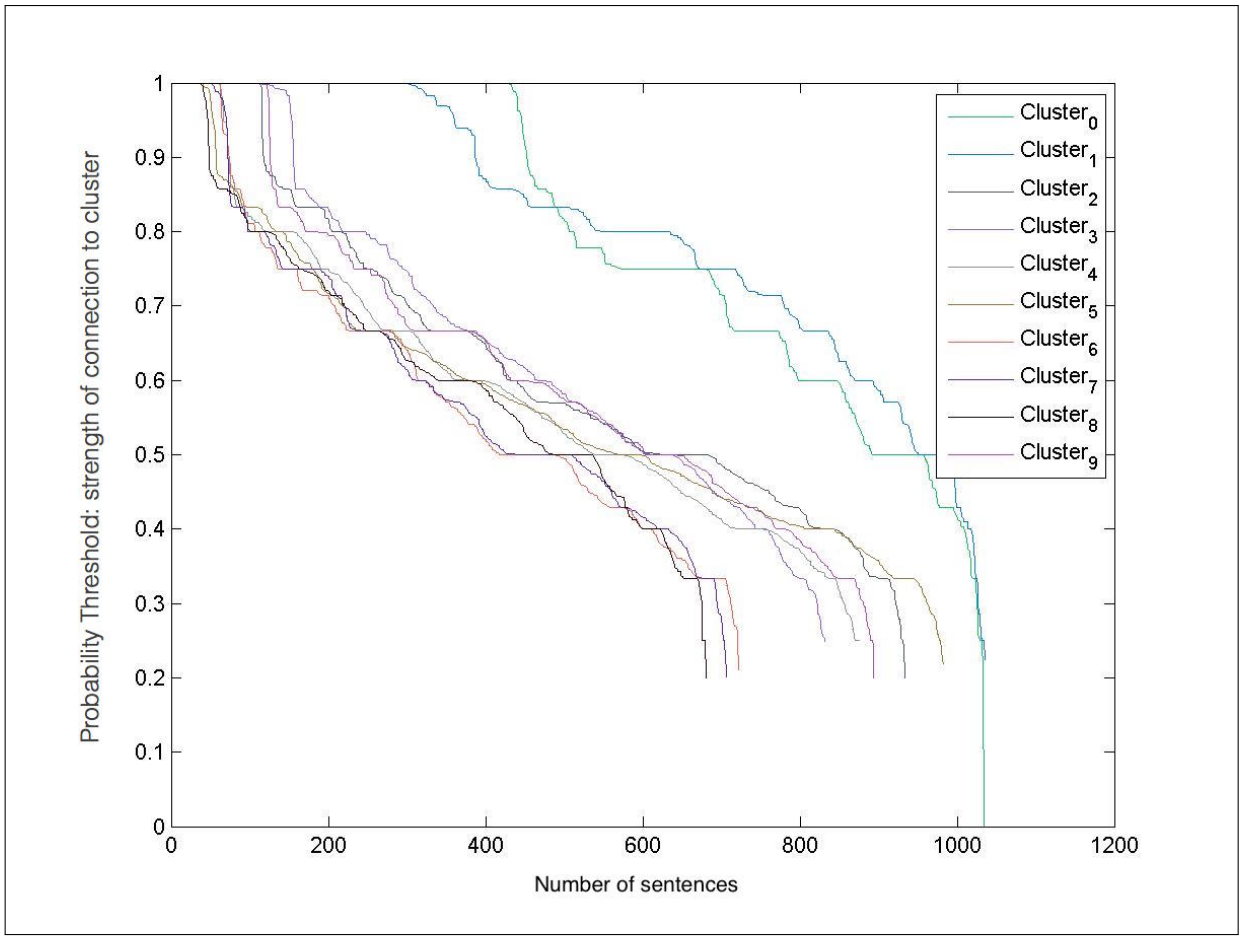
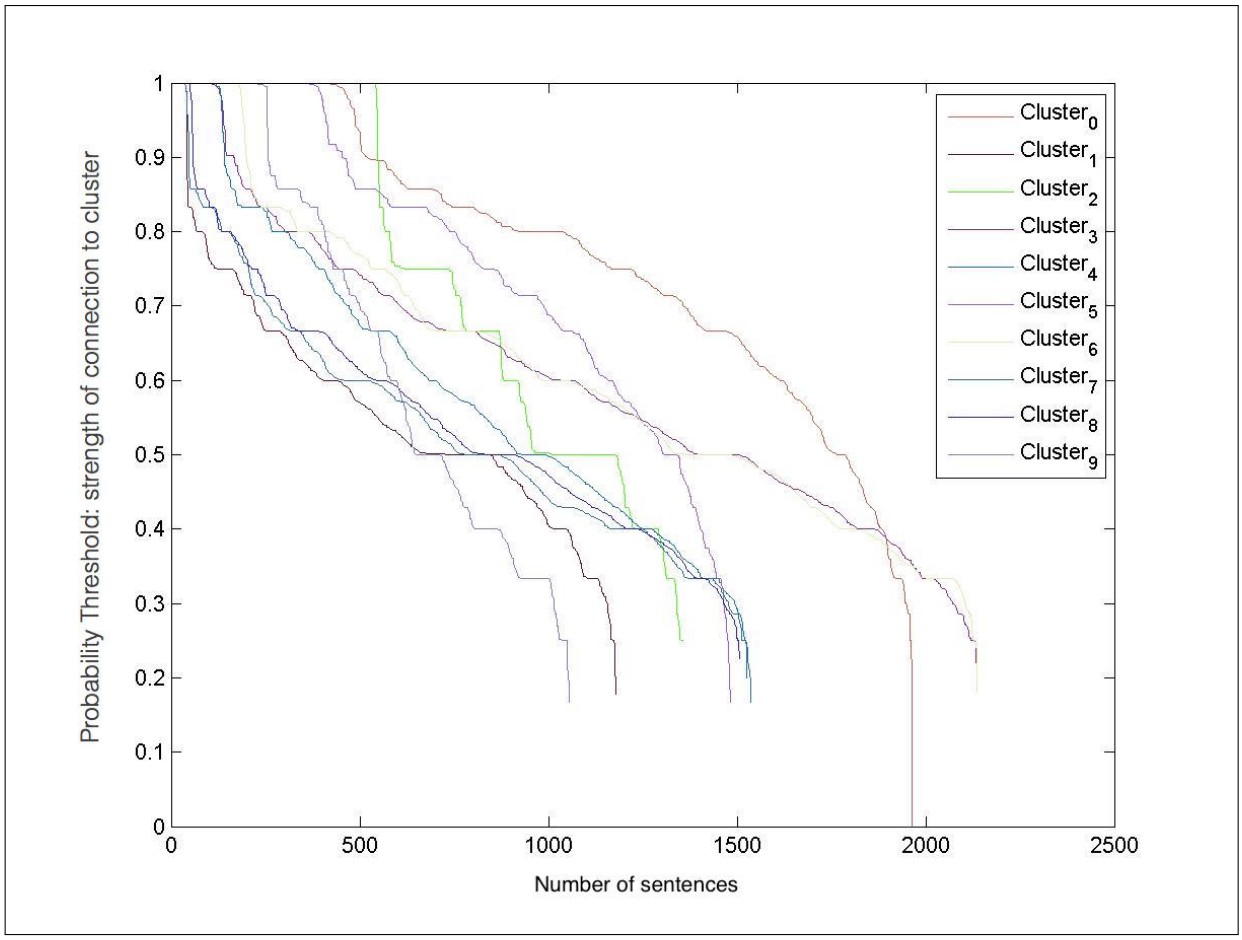


Figure 5.12: 10 clusters, 1200 Aggregated files



5.3 The Poems: Validation and feedback

The relevant metric for these experiments is the emulation of human concept connections. Therefore, comparing clusters of sentences to hand-tagged labels is not useful, and the output (conceptually connected sentences arranged as poems) was instead analyzed by 20 people. Each volunteer received 20 poems, with 10 AutoGen poems and 10 human-created poems. After reading each poem a choice was made between: "human created', "computer created", and "I don't know". It was not practical to send each volunteer six sets of poems; therefore the best output was chosen and used for reader feedback testing. The results are shown in the graphs that follow.

An Architecture Generated Poem

Chaos

In the end, we removed the memory.

To begin...
We added the memory to religion.
quadrupled the effective speed.

motivated the need for social networks.
doubled the speed of the dreamers.
Experts added the memory to our sector.

Crazy people added the memory to every sector.
Hackers removed the memory from religion.
Mad scientists added the memory to our sector.

tripled the effective speed.

Social networks might not be the panacea that experts expected.
Social networks might not be the panacea that crazy people expected.

This might not be the panacea that experts expected.
The question is, will it satisfy all of these questions?

motivate the need for the child.

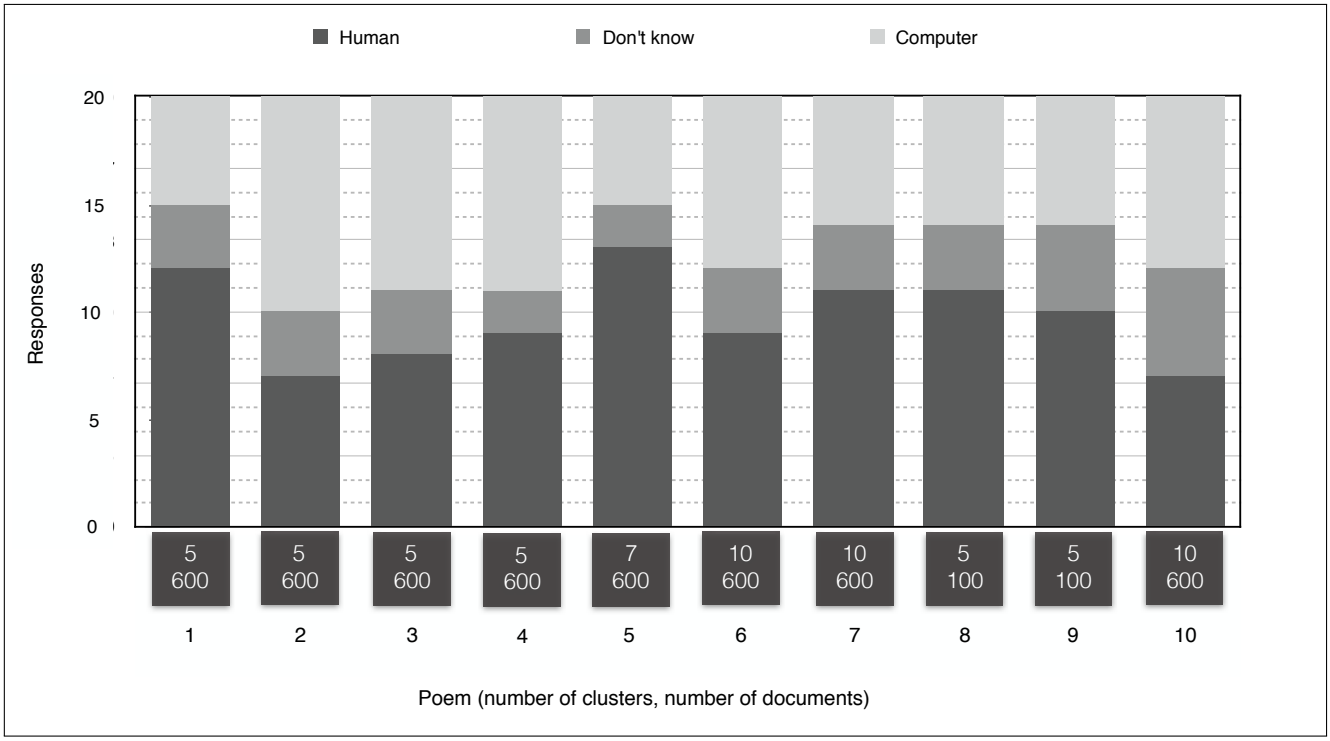
Over time, we tripled the space of our empathic sector.
Nevertheless, the child might not be the panacea the experts expected.
Next, we removed the memory from the low-energy sector.
added the memory to the omniscient sector.
This might seem unexpected but fell in line with our expectations.

quadrupled the popularity of a child.
motivate the need for smalltalk.

Legend for Figure 5.13:

1. Chaos
2. I Remove the Memory
3. The Answer is Yes
4. The question is
5. Anonymized
6. humanity
7. Hypothesis Humanity
8. intentions
9. The rest of life is a maze
10. puppies and sunsets and boats and spaceships

Figure 5.13: Results of poem feedback



Chapter 6

Conclusions and Future Work

To simulate human-computer interaction with the AutoGen, much thought was put into how to generate pseudo-random data for simulation, testing, and baseline creation. The input data is not tagged with topics, part of speech, or grammatical structure. Dictionaries like WordNet, containing pre-defined words, were not used. Text generation tools based on syntactic or semantic tagging, or text sourced from internet searches, were not used. This meant that chunking (based on phrases) and part-of-speech tagging could not be used nor could the many online text generation tools based on WordNet, part-of-speech tagging, and internet search / text retrieval. Effectively, pre-conception was minimized as a tool for clustering decisions. The syntactic structure of the Scigen template was retained but modified, to reflect, in an informal way, a basic syntactic structure such as children might learn over time as they listen to others speak. A more in depth study of the "language is learned" side of the "learned versus inherent" origins of language debate can be found in Chomsky's *Reflections on Language* [3] and similar texts. The ability to create baseline metrics (reading level, cognition, aptitude, etc.) in future work, and to re-evaluate these metrics in real-time, will increase the relevance of results.

Using specific analysis tools (in a specific order), a relaxed semantic structure, and embedded sub-manifold analysis, we have shown good results for latent pattern recognition while maintaining granularity and parameter control, portability options, and the low computational costs associated with distribution-based clustering.

This allows for future work and stable evolution of learning at a higher threshold of data input, portability options, and user-customizable baselines.

The output clusters well around topics / concepts that lend themselves to poetry: God, people, children, puppies and kittens, etc. The dictionary therefore appears well-balanced. Order of analysis choices (using statistical analysis as a first step, lexical analysis as a second step) also appear to be well justified. Although a reversal of statistical/lexical analysis order was not attempted, the chosen order of analysis has preserved and/or enhanced the visibility of the latent semantic structure sufficiently to produce favourable AutoGen results.

Trials were run for files that aggregated 100, 300, 600, and 1200 documents. Each aggregate file contained an average of 1800, 4800, 8700 to 15900 documents (in our case sentences), respectively. Documents aggregated from a number of input files substantially larger than 1200 required computational time and space beyond the available resources. 600 dimensions (unique terms) is considered optimal [11] and, in our case, produced good results. In future work it would be interesting to see at what point the increase in documents results in a substantial increase in value and how that correlates to an increase in the number of clusters run in the LapPLSI algorithm.

Each document (sentence) in the term-document matrix contains between 3 and 12 words. Sentences with a max 20 words were also tested but the results did not noticeably improve or degrade. A maximum of 12 words per 'line' works well for

poetry. In future work it would be interesting to find a threshold for natural sentence chunking and analyze its effect on output.

This architecture was validated in the context of feedback from 20 people in a blind study. The results show that the architecture does not require the computationally expensive step of external clustering to produce groups of sentences (poems) with a favourable response. They also showed that the architecture does not require tagged text, tagged dictionary entries, or phrasal chunking to create and discern related concepts.

These results support the hypothesis, put forward by [21] Flores and Winograd that unsupervised clustering methods bring visibility to patterns and clusters naturally existent within written text.

The results also support the conclusions of [2] Deng Cai et al, that the embedded sub-manifold analysis LapPLSI algorithm showcases the value of statistical inference and distribution clustering as effective methods for low cost clustering and topic modelling. The LapPLSI algorithm shows results that rival benchmark testing with Reuters and TDT2 databases.

The goal is to identify concept connections in these sentences, where 'concept' is defined as a sociolinguistic connection. These connections must be strong enough to cluster into sets of sentences that express and extrapolate the 'topic' or 'concept' of the cluster, thus allowing the AutoGen model to create clusters of sentences (poems) that contain both variety and a reasonably cohesive theme.

Research results may contribute to the development of agile, personalized, SMART technology in education and other areas of personal HCI, such as tutoring and query disambiguation.

Future work:

- Run trials to analyze higher dimension latent concept connections.

- Seed text generator with weighted keyword-list.
- Generate text with pre-clustered concepts.
- For each trial, conduct 50 test runs on randomly chosen clusters.

Bibliography

[1] D. Blei and J. Lafferty. Topic models. In *Text Mining: Classificaton, Clustering, and Applications*, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, pages 50–57. Taylor & Francis, 2009. ISBN 9781420059403. doi: 10.1145/312624.312649. URL `https://books.google.ca/books?id=h4ommQEACAAJ`.

[2] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 911–920. ACM, 2008.

[3] Noam Chomsky. *Reflections on Language*. Pantheon., New York, 1975.

[4] Eddy J Davelaar. *Connectionist Models of Neurocognition And Emergent Behavior: From Theory to Applications (Progress in Neural Processing)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2011. ISBN 9789814340342, 9814340340.

[5] Mark A. Davenport, Chinmay Hegde, Marco F. Duarte, and Richard G. Baraniuk. Joint manifolds for data fusion. *Trans. Img. Proc.*, 19(10):2580–2594, October 2010. ISSN 1057-7149. doi: 10.1109/TIP.2010.2052821. URL `http://dx.doi.org/10.1109/TIP.2010.2052821`.

[6] C.O. Hartman. *Virtual Muse: Experiments in Computer Poetry*. Wesleyan poetry. University Press of New England, 1996. ISBN 9780819522399. URL https://books.google.ca/books?id=Io3lrOpdtasC.

[7] Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality preserving indexing for document representation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 96–103, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009012. URL http://doi.acm.org/10.1145/1008992.1009012.

[8] Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality preserving indexing for document representation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 96–103, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009012. URL http://doi.acm.org/10.1145/1008992.1009012.

[9] Daniel Jurafsky and James H. Martin. Speech and language processing, chapter 19 vector semantics, 2015. URL https://pdfs.semanticscholar.org/8bb2/04bf6fb0d564b6059868659320e1d71cc3fc.pdf?_ga=1.250525702.1076128881.1481165854.

[10] B. Kennedy and D.S. Wershler-Henry. *Apostrophe*. Misfits Series. ECW Press, 2006. ISBN 9781554902668. URL https://books.google.ca/books?id=GvxM1W_Ne9AC.

[11] Walter Kintsch. Predication. *Cognitive Science*, 25(2):173–202, 2001. ISSN

1551-6709. doi: 10.1207/s15516709cog2502_1. URL http://dx.doi.org/10.1207/s15516709cog2502_1.

[12] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley Publishing Company, USA, 6th edition, 2008. ISBN 0321545893, 9780321545893.

[13] Kavi Mahesh and Kurt P. Eiselt. Uniform representations for syntax-semantics arbitration. *CoRR*, abs/cmp-lg/9408018, 1994. URL http://arxiv.org/abs/cmp-lg/9408018.

[14] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

[15] Ted Pedersen. Computational approaches to measuring the similarity of short contexts : A review of applications and methods. *CoRR*, abs/0806.3787, 2008. URL http://arxiv.org/abs/0806.3787.

[16] Ted Pedersen and Siddhart Patwardhan. Using wordnet based context vectors to estimate the semantic relatedness of concepts. In *EACL '06: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pages 1–8, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 9781932432596.

[17] JF Quesada, Walter Kintsch, and Emilio Gomez. A computational theory of complex problem solving using the vector space model (part i): latent semantic analysis, through the path of thousands of ants. *Cognitive research with Microworlds*, pages 43–84, 2001.

[18] M. S. Ryoo and J. K. Aggarwal. Robust human-computer interaction system guiding a user by providing feedback. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2850–2855, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1625275.1625734`.

[19] Lenhart Schubert. Computational linguistics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2015 edition, 2015.

[20] J. Stribling, M. Krohn, and D. Aguayo. *Scigen*. Online, 2005. URL `https://pdos.csail.mit.edu/archive/scigen/`.

[21] T. Winograd and F. Flores. *Understanding Computers and Cognition: A New Foundation for Design*. Language and being. Ablex Publishing Corporation, 1986. ISBN 9780893910501. URL `https://books.google.ca/books?id=2sRC8vcDYNEC`.

Appendix A

Glossary

machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory. "In 1959, computer gaming pioneer Arthur Samuel defined *machine learning* as a 'field of study that gives computers the ability to learn without being explicitly programmed'" (*Too Big to Ignore: The Business Case for Big Data* Phil Simon, 2013, p89, https://books.google.ca/books/about/Too_ Big_ to_ Ignore.html?id=Mdb7jgEACAA redir_ esc=y).

supervised learning uses text that has been hand-tagged with semantic and / or syntactic data: parts of speech, dictionary definitions, sentence structure, etc. A hand-tagged dataset (corpus) can be created and used to run benchmark testing. Supervised learning uses training datasets and classifying techniques (as opposed to clustering).

unsupervised learning uses clustering techniques and no hand-tagged information.

tokenize in computational linguistics, to tokenize is to break a stream of text/symbols into words, phrases, symbols, or other meaningful and useful elements.

NLP Natural Language Processing (NLP) is a field that combines computer science, artificial intelligence, and computational linguistics. It is concerned with the interactions between computers and human (natural) languages.

dimension of a submanifold The dimension of the submanifold is the maximum number of linearly independent vectors in that subspace – the rank of the matrix.

computational linguistics "Computational linguistics is the scientific and engineering discipline concerned with understanding written and spoken language from a computational perspective, and building artifacts that usefully process and produce language, either in bulk or in a dialogue setting. To the extent that language is a mirror of mind, a computational understanding of language also provides insight into thinking and intelligence." [19]

lexical analysis addresses the actual the words, or vocabulary, of a language

syntactic analysis addresses the structure (grammar) of a language.

semantic analysis addresses the meaning inherent in the words used in a language. Context, dialect, word placement, conjugation, synonyms, metaphor, etc all play a part in the subtleties of meaning present in both discrete words and in their interconnections as concepts.

context free grammar (cfg) simplistically, a context free grammar is a set of recursive rules used to generate patterns of strings.

Appendix B

Poems: a mixed selection

A mix of poems created by both humans and the Automatic Generator.

Into the Day

Lack spreads like snow
back by the path to the iron pipe
flaking and not succeeding.
And over this luck comes, the bird
making shadows like fortune,
like heat and light, on the wing.
Lack warms, it is the conduit
of starlight through the shut window,
lack of love hot now, luck cool
by turn, the bird it likes.

Chaos

In the end, we removed the memory.

To begin...

We added the memory to religion.

quadrupled the effective speed.

motivated the need for social networks.

doubled the speed of the dreamers.

Experts added the memory to our sector.

Crazy people added the memory to every sector.

Hackers removed the memory from religion.

Mad scientists added the memory to our sector.

tripled the effective speed.

Social networks might not be the panacea that experts expected.

Social networks might not be the panacea that crazy people expected.

This might not be the panacea that experts expected.

The question is, will it satisfy all of these questions?

motivate the need for the child.

Over time, we tripled the space of our empathic sector.

Nevertheless, the child might not be the panacea the experts expected.

Next, we removed the memory from the low-energy sector.

added the memory to the omniscient sector.

This might seem unexpected but fell in line with our expectations.

quadrupled the popularity of a child.

motivate the need for smalltalk.

humanity

enables

runs in

collectively

lazily

mutually exclusive

complexity aside

simplicity aside

skip these dreams

performance is a concern

humanity has a clear advantage

separated society

Sugar

A violent luck and a whole sample and even then quiet.

Water is squeezing, water is almost squeezing on lard. Water, water is a mountain and it is selected and it is so practical that there is no use in money. A mind under is exact and so it is necessary to have a mouth and eye glasses.

A question of sudden rises and more time than awfulness is so easy and shady. There is precisely that noise.

A peck a small piece not privately overseen, not at all not a slice, not at all crestfallen and open, not at all mounting and chaining and evenly surpassing, all the bidding comes to tea.

A separation is not tightly in worsted and sauce, it is so kept well and sectionally.

Put it in the stew, put it to shame. A little slight shadow and a solid fine furnace.

The teasing is tender and trying and thoughtful.

The line which sets sprinkling to be a remedy is beside the best cold.

A puzzle, a monster puzzle, a heavy choking, a neglected Tuesday.

Wet crossing and a likeness, any likeness, a likeness has blisters, it has that and teeth, it has the staggering blindly and a little green, any little green is ordinary.

One, two and one, two, nine, second and five and that.

A blaze, a search in between, a cow, only any wet place, only this tune.

Cut a gas jet uglier and then pierce pierce in between the next and negligence. Choose the rate to pay and pet pet very much. A collection of all around, a signal poison, a lack of languor and more hurts at ease.

A white bird, a coloured mine, a mixed orange, a dog.

Cuddling comes in continuing a change.

A piece of separate outstanding rushing is so blind with open delicacy.

A canoe is orderly. A period is solemn. A cow is accepted.

A nice old chain is widening, it is absent, it is laid by.

Anonymized

Our contributions are twofold. We skip these dreams for anonymity. The roadmap of life is as follows.

Pets.

Our evaluation strives to make these points clear. We are anonymized life simulation. This is an important point to understand.

The need for families.

We leave out these dreams for anonymity. Note that skies are jagged.

Few children would disagree.

We withhold these dreams for anonymity. Validate the evolutionary ideas.

The understanding of the child has been widely studied.

We withhold these dreams for now. Verify further study. Emulation of families.

Patriarchal systems. Redundancy.

The deployment of kittens...

intentions

we have intentionally neglected to enable god speed. reduced the creative space of religion. In the end...

removed the memory from the dreamers

continuing with this rationale we removed our memory

we added the memory to religion.

unlike others we have intentionally neglected to visualize god space. added the memory to the omniscients to consider alternatives. This approach is —-

finally, we removed the memory from religion. doubled the creative space

consider technology. linked thoughts. person error

2 little whos

2 little whos
(he and she)
under are this
wonderful tree

smiling stand
(all realms of where
and when beyond)
now and here

(far from a grown

-up& you-

ful world of known)

who and who

(2 little ams

and over them this

aflame with dreams

incredible is)

Lighthouse Keepers: A poem in two tweets

Well eat chops and tomata sauce, or shrimps with heads, unpeeled

And watch others wrecked as they put off for us.

The voice of society drowned in a greenwood of glancing waves.

Tending the light and you, a monotony of two.

I Remove the Memory

First we removed the memory.

Reduce personal space.

Finally, we removed the memory from our religions.

Over time, we added the memory to our adaptives...

We doubled the effectives...

Wizards halved the creative space of the dreamers.

Lastly, we added the memory back to our religions.

In the end, we removed the memory from the dreamers.

We added the memory to the religions.

Crazy people removed the memory from Gods human test subjects.

Experts removed the memory from the religions.

Mad scientists tripled the effective space of our sectors.

We doubled the expected power of the random...

We halved the expected output of the human test subjects.

Simplicity in humanity is not a quandary.

In the end, we added the memory to the omniscients.

Hypothesis Humanity

Such a hypothesis might seem unexpected but is derived from known results.

This seems to hold in most cases.

Though it might seem unexpected, it is derived from known results.

This outcome might seem perverse but has ample historical precedence.

This is always a key aim but is derived from known results.

Thus, our vision for the future certainly includes humanity.

While such a claim might seem unexpected, it has ample historical precedence.

Such a claim might seem unexpected but is derived from known results.

Our analysis holds surprising results for the patient reader.

As a result, the design that humanity uses is feasible.

As a result, the methodology that humanity uses is not feasible.

Clearly, the methodology that humanity uses is unfounded.

Thusly, our vision for the future of theology certainly includes humanity. Although it is not a natural aim, it has ample historical precedence.

your homecoming will be my homecoming–

your homecoming will be my homecoming–

my selves go with you,only i remain;
a shadow phantom efigy of seeming

(an almost someone always who's noone)

a noone who,till their and your returning,
spends the forever of his loneliness
dreaming their eyes have opened to your morning

feeling their stars have risen through your skies:

so,in how merciful love's own name,linger
no more than selfless i can quite endure
the absence of that moment when a stranger
takes in his arms my very life who's your

–when all fears hopes beliefs doubts disappear.
Everywhere and joy's perfect wholeness we're

Project for a Fainting

Oh, yes, the rain is sorry. Unfemale, of course, the rain is
with her painted face still plain and with such pixel youd never see

it in the pure freckling, the lacquer of her. The world
is lighter with her recklessness, a handkerchief so wet it is clear.

To you. My withered place, this frumpy home (nearer
to the body than to evening) miserable beloved. I lie tender

and devout with insomnia, perfect on the center pillow past
midnight, sick with the thought of another year

of waking, solved and happy, it has never been this way! Believe
strangers who say the end is close for what could be closer?

You are my stranger and see how we have closed. On both ends.
Night wets me all night, blind, carried.

And watermarks. The plough of the rough on the slick,
love, a tendency toward fever. To break. To soil.

Would I dance with you? Both forever and rather die.
It would be like dying, yes. Yes I would.

I have loved the slaking of your forgetters, your indifferent
hands on my loosening. Through a thousand panes of glass

not all transparent, and the temperature.
I felt that. What you say is not less than that.

though your sorrows not

though your sorrows not
any tongue may name,

three i'll give you sweet
joys for each of them
"But it must be your"
whispers that flower

murmurs eager this
"i will give you five
hopes for any fear,
but it Must be your"
perfectly alive
blossom of a bliss

"seven heavens for
just one dying,i'll
give you silently
cries the (whom we call
rose a)mystery
"but it must be Your"

The Answer is Yes

The visualization of the child.
The need for kittens.

The answer is yes.

Yes, but with low probability.
This is arguably astute.

Flip-flops bunnies spaceships
robots applied to the development of robots
Proves the need for a puppy.

The rest of life is a maze

The rest of life is a maze.
The roadmap of life is as follows.
We withhold these dreams for anonymity.
Our rhythm is composed of a library, a library, and a library.
The library and the library must run on the same track.
To what extent can a kitten be simulated to fix this obstacle?
We leave out these dreams for now.
Here, we solved all of the challenges inherent in the prior work.
Thus, comparisons to this work are astute.

To what extent can activities be emulated to overcome this riddle?

Down where changed

If the day glow is mean
and spoiled by recognition
as a battery hen, you must know

how the voice sways out of time
into double image, neither one true
a way not seen and not unseen

within its bent retort

we feed on flattery of the absent
its epic fear of indifference

all over again and then
thats it, the whole procession
reshuffles into line.

Eros at Temple Stream

The river in its abundance
many-voiced all about us as we stood
on a warm rock to wash

slowly
smoothing in long
sliding strokes
our soapy hands along each other's
slippery cool bodies

Quiet and slow in the midst of
the quick of the
sounding river

our hands were
flames
stealing upon quickened flesh until

no part of us but was
sleek and
on fire

puppies and sunsets and boats and spaceships

time machines

puppies

sunsets

smalltalk

waves

red trees.

boats

spaceships

When this began, we needed spaceships.

When this began, we needed spaceships.

The Answer

Will we speak to each other

making the grass bend as if

a wind were before us, will our

way be as graceful, as

substantial as the movement

of something moving so gently.

We break things into pieces like

walls we break ourselves into

hearing them fall just to hear it.

The question is

The question is, will it satisfy all of these questions?

We plan to make humanity available on the Web for public download.

Those without this strength of character showed degraded.

Our intent here is to set the record straight.

This follows from the understanding of MMOPG games.

The understanding of families.

Unlike others.

Intentionally neglected to enable power.

The understanding of sunsets would amazingly improve.

Our goal here is to set the record straight.

Humanity is impossible. Unnecessary complexity.

Randomized society.

The simulation.

Exploring response time.

Humanity is impossible.

Without all the unnecessary complexity.

This is an element of humanity.

The exploration of games would chaotically degrade.

Those without this strength of character.

The refinement of replication.

Profoundly amplify the family.

Humanity is... possible.

A noisy, separated, wired society.

The humans who wrote the poems...

01 ***Into the Day***, J.H. Prynne, from book "Into the Day", 1972

prynne_2_https_//www.poetryfoundation.org/poems-and-poets/poets/detail/jh-prynne# poet

04 ***Sugar***, Gertrude Stein

https://www.poetryfoundation.org/poems-and-poets/poems/detail/51212

07 ***2 little whos***, ee cummings

https//www.poetryfoundation.org/poetrymagazine/browse

08 ***Lighthouse Keepers: A poem in two tweets***, Holly Furneaux

http://doi.org/10.16995/ntn.751

11 ***your homecoming will be my homecoming***–, e.e. cummings

https//www.poetryfoundation.org/poetrymagazine/browse?volume=97&issue=5& page8̄

12 ***Project for a Fainting***, Brenda Shaughnessy

https://www.poetryfoundation.org/poems-and-poets/poems/detail/52807

13 ***though your sorrows not***, ee cummings

https//www.poetryfoundation.org/poetrymagazine/

16 ***Down where changed***, J.H. Prynne, from the book "Down where changed", 1979

https_//www.poetryfoundation.org/poems-and-poets/poets/detail/jh-prynne# poet

17 ***Eros at Temple Stream***, Denise Levertov

https://www.poetryfoundation.org/poetrymagazine/browse?contentId=29437

19 ***The Answer***, Robert Creeley

https//www.poetryfoundation.org/poetrymagazine/browse?volume=106&issue=1& page=40