UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Generalized optimization models of linguistic laws

David Carrera Casado

Director
Ramon Ferrer Cancho
(Department director)

(Data defensa)

MASTER IN INNOVATION AND RESEARCH IN INFORMATICS
Computer Networks and Distributed Systems

Abstract

Quantitative linguistics studies human language using statistical methods. Its aim is to build general theory from the statistical laws observed in a wide variety of languages. As part of the scientific method, this theory should be able to make novel predictions. This thesis is based on a family of models of human language ([12] and generalized in [14]). These models have shown to reproduce language laws, such as Zipf's law. They have also been used to make predictions, such as predicting the biases present in child word learning. This family of models is based on the minimization of a cost function. The cost function is defined using a combination of information theoretic measures on a bipartite graph of associations between words (or, more generally, forms) and meanings (more generally, counterparts). It balances between the entropy of words and the mutual information of words and meanings. Entropy is a measure of surprisal, the cost of the speaker, and should be minimized. Mutual information is the amount of information obtained from a meaning while observing a word, the cost of the listener, and it should be maximized. The model is then optimized with a Markov Chain Monte Carlo method at zero temperature. This thesis is centered on two models belonging to this family, the internal model and the external model. This thesis makes several contributions in relation to these models. The mathematical equations defining them are derived, including dynamic equations which reduce the computational complexity of the optimization process. In addition, several techniques are introduced which aim to reduce the significant problem of numerical error due to floating point arithmetic without compromising efficiency. Another contribution is the replication of results obtained by previous models based on this family which had been published originally with replicability issues. The models are simulated and the linguistic laws they can predict and to which degree are examined. A key contribution is that these models are able predict the relationship between the age of a word and its frequency. This prediction is robust and appears in all cases with any combinations of parameters. Finally, a tool has been developed and released as open source with the aim that others can easily replicate these results and investigate other properties of this family of models.

Contents

Chapter 1

Introduction

Human languages obey many regularities which have been studied and formalized into laws. One of the most well known such laws is Zipf's law, which states that

$$p(i) \sim i^{-\alpha} \tag{1.1}$$

where i is a word's rank. Generally, $\alpha \approx 1$. [23] This is the case not only for English but for every tested human language. [15] Figure 1.1 shows this relationship for the first 10 million words of Wikipedia dumps in 30 different languages. This relationship also holds for artifical languages like Esperanto. There have been many efforts to understand why this and other empiric laws occur consistently in human language.

This thesis focuses on a family of models [12] introduced initially to shed light into the origins of Zipf's law. [10] [13] The family of models aims to explain language laws such as Zipf's law from the association of words and meanings. The focus of this thesis are two models from this family, which were generalized with the addition of a new parameter ϕ. [14] They are referred to as the "internal model" and the "external model" throughout. The key difference between these two models is whether they consider meanings to follow a probability distribution *internal* or *external* to the model. In the internal model the probability of a meaning is determined entirely by the connections between words and meanings. The external model determines the probability of a meaning in great part from an external *a priori* probability distribution. This *a priori* probability can be considered the probability that the element referred to by the meaning is found in nature. As an example, if elephants are rare but dogs are plentiful, the meaning "elephant" should be rarer than the meaning "dog".

In previous work [13] [10] Zipf's law of word frequencies has been predicted by the two models that are the focus of this thesis. Just a few paragraphs ago this document opened with its formulation in Equation (1.1), the relationship between a word's frequency and frequency rank follows a power law. But there exist other models that can explain language laws. A popular explanation for Zipf's law of word frequencies is the random typing model.[5] As a short summary, this model states that if characters are typed in sequence with a random

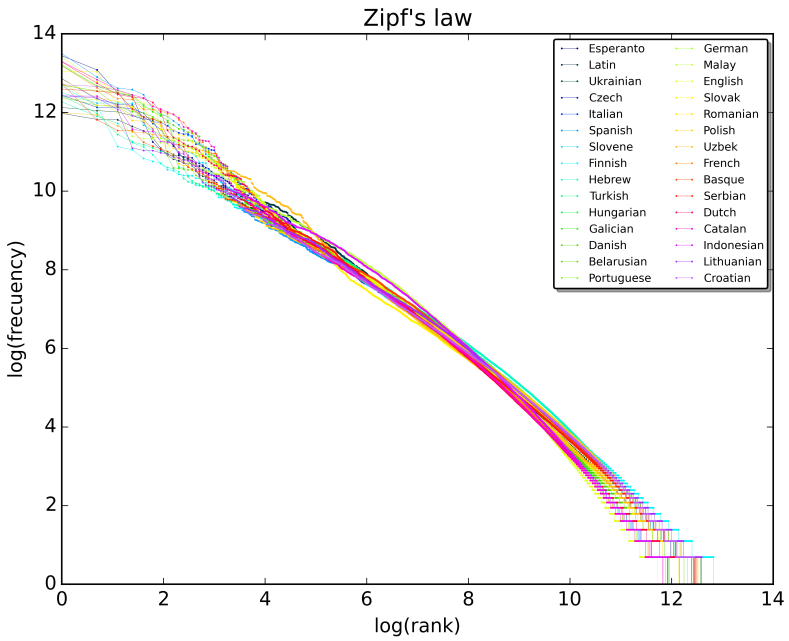


Figure 1.1: The relationship of the logarithm of a word's frequency versus its rank for 30 human languages including Esperanto, an artifical language. Data from 30 Wikipedia dumps from October 2015.
(Sergio Jimenez/Wikimedia, CC BY-SA 4.0 [20])

chance of typing in a word separator then Zipf's law is reproduced. There are a few issues with Miller's model, such as the assumption that words are independent from each other. Random typing is often offered as a null hypothesis to the theory that Zipf's law comes from the *principle of least effort* but even that has been called into question as it can also be considered a form of optimal coding. [11]

But Zipf's law for word frequencies is not the only linguistic law. Zipf's meaning frequency law is not as popular as the word frequency law. It states that the relationship between the number of meanings of a word, μ, and its frequency, f, is [23]

$$\mu \propto f^{\delta}.$$

Zipf derived the meaning-frequency law from his law of word-frequency, Equation (1.1) and from his law of meaning distribution,

$$\mu \propto i^{-\gamma}$$

where i is the rank of the word.

As with the constant α, δ and γ can be estimated using a regression method from data.

Zipf inferred $\delta \approx 1/2$ from $\gamma \approx 1/2$ and $\alpha = 1$. Later on, others [8] have shown the relationship

$$\delta = \frac{\gamma}{\alpha}. \tag{1.2}$$

between these exponents.

Miller's random typing model is not able to predict this other linguistic law as the meanings of the words are not taken into account. It is also impossible for it to predict other more complex phenomena such as the vocabulary learning bias in children. There are very good arguments [14] in favor of this family of models being able to predict this law and at least the internal model has been able to predict vocabulary learning biases. Both the case $\phi = 0$ [9] and for the more generic formulation with any value of ϕ [4].

In his book [23] Zipf argued also that older words (words that have existed in a language for a longer amount of time) would be more frequent. This was tested empirically, as seen in Figure 1.2.

While this relationship between word frequencies and word ages could also not be predicted by random typing, there is another alternative model that can predict it.

Simon's model [21] model argues that power laws appear as a result of the way the system is formed. In the case of language, constant addition of new words and addition of new instances of already existing words at a rate proportional to the number of instances of a word Here, time is a factor. And intuitively it seems that older words would indeed be more frequent. However, Simon's model cannot explain the meaning distribution law as it does not take meaning into account and lacks the complexity to make predictions such as the vocabulary learning biases of children.

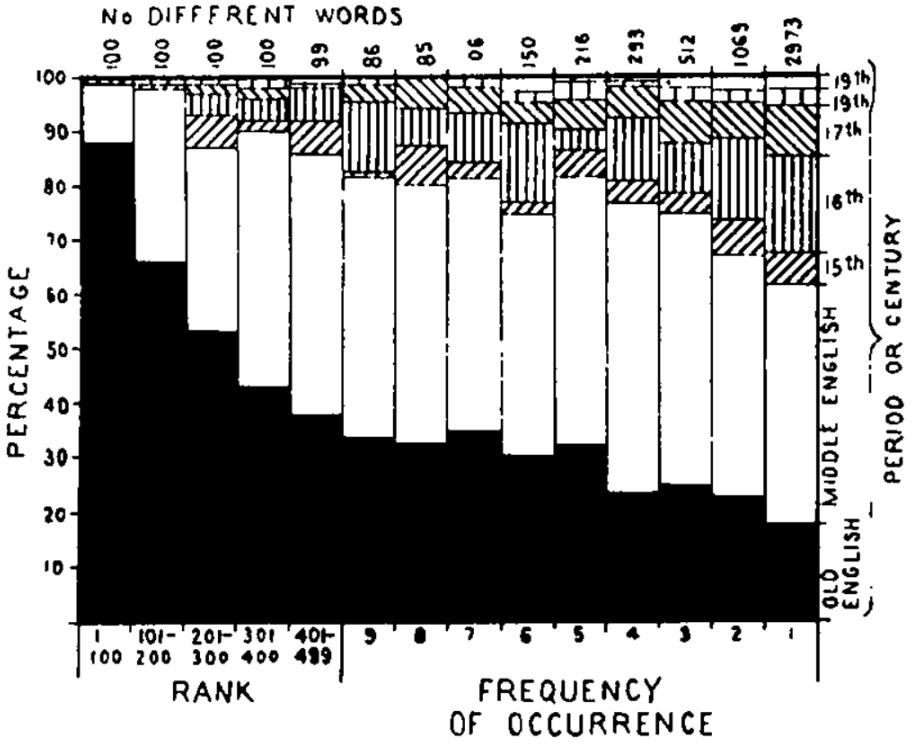


Figure 1.2: Age of a word as a function of its frequency from Zipf's famous book. [23] The right side of the y axis indicates the historical period or century when a word was introduced and the left side the percentage of words. Each of the colors and patterns of the columns in the graph correspond with a time period. As indicated in the x axis, the first five columns refer to the most common words by rank, while the next columns refer to the words with the specified frequency of occurrence.

	Random Typing	Simon's Model	Internal model ($\phi = 0$)	External model ($\phi = 0$)	Internal model ($\phi \neq 0$)	External model ($\phi \neq 0$)
Rank-frequency law	Yes	Yes	Yes	Yes	No	Yes
Meaning distribution law	No	No	Yes	No	No	Yes
Age-frequency law	No	Yes	Yes	Yes	Yes	Yes
Vocabulary learning bias	No	No	Yes [9]	Unknown	Yes [4]	Unknown

Table 1.1: A summary table comparing various models of human language. Columns show various models of human language: Random Typing and Simon's model. Then the two models studied in this thesis: internal and external model. The value of ϕ indicates whether it is the older version of the model (equivalent of the general model with $\phi = 0$) or the more general version with $\phi \neq 0$. Rows show various predictions that these models could or could not do. The vocabulary learning bias was shown to be predicted with the internal model and more recently with the more generic version of the model ($\phi \neq 0$)

Table 1.1 shows a comparison between the two models seen here (random typing and Simon's model) and the two models that this thesis presents. The two models are further fleshed out in following sections.

Unlike Miller's and Simon's models, these models are based on the argument that human language is result of attempting to minimize the effort of both the speaker and the hearer. [13] [23] Something that Zipf referred to as the *principle of least effort*. That is why these models are based on the minimization of a cost function, which is calculated in terms of information theoretic measures.

This approach to optimization in human languages is not new. Terry Regier argued [18] that color naming in human language corresponded with optimal partitions of the color space. Regier's research showed that, indeed, color naming in many human languages was closely related to the optimal partitions of the color space. The idea of language being "optimal" is not new.

From the computational point of view, an important aspect of the optimization process is the evaluation of the cost function. The cost function can be evaluated *statically*, recalculating it completely from scratch each time a change (or mutation) is made to the underlying graph. However, it is possible to calculate only the changes that take place due to this mutation. In this way the calculation is computationally less complex. This is not a new approach, in shortest path calculation algorithms, when a change takes place it is common to adjust the result based on the change instead of recalculating the entire process. [3] The dynamic approach offers a significant speedup that justifies the issues that arise as a consequence, such as the increase in mathematical complexity and introduction of greater numerical error.

The tool developed to study these models is released under an open source license. It is released with the hope that it will be useful to replicate these results, and also that it can be used to study similar models.

Now follows an introduction to the model. It is not as in depth as what can be found in following chapters, but it does give the basic mathematical points of the models.

1.1 Introduction to the model

This is a short introduction to the models presented in this thesis. In this section, mathematical definitions and notation common to both models is given. They will be used throughout the remainder of the thesis and specially throughout Chapter 3 where a full explanation of both models is given.

Section 1.1.1 presents the bipartite graphs that form the *skeleton* of these models. Section 1.1.2 presents the information theoretic aspects, common to both models. In Section 1.1.3 the role of the ϕ parameter is explained. Sections 1.1.4 and 1.1.5 cover the parts that unique to the internal and external model respectively. These are the *flesh*, each covering the *skeleton* in a different way. Section 1.1.6 gives an overview of the optimization process. Section 1.1.7 explains the reasoning and difference of the static and dynamic versions of the implemented algorithms.

1.1.1 Bipartite graph

Both models studied in this thesis are based on the idea of a bipartite graph. Bipartite graphs are comprised of two sets of elements and edges can only appear between an element of one set and an element of the other.

Our two sets are S and R. S is a set of size n containing all words. The notation s_i is used to refer to some element i of the set S. R is a set of size m containing all meanings. The notation r_j is used to refer to some element j of the set R.

The bipartite graph is represented using the adjacency matrix $A_{n,m}$ (or simply A) in most cases. $A_{n,m}$ is a $n \times m$ binary matrix representing whether an edge exists or not in the graph. Mathematically, each element of the matrix A is defined as

$$a_{i,j} = \begin{cases} 1 & \text{if there exists an edge between } s_i \in S \text{ and } r_j \in R \\ 0 & \text{otherwise.} \end{cases}$$

In some cases it is more convenient to represent the bipartite graph as the set E of all edges,

$$E = \{\, (s_i, r_j) \mid \text{there exists an edge between } s_i \in S \text{ and } r_j \in R \,\}.$$

For brevity, the pair (s_i, r_j) is often represented as (i, j).

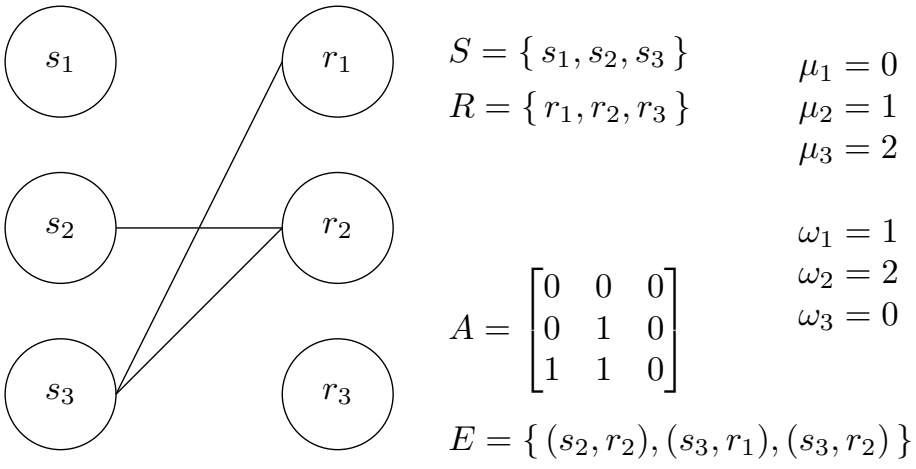


Figure 1.3: A bipartite graph with the corresponding adjacency matrix A, set of edges E and vertex degrees μ and ω for each vertex.

The degree of the word i is given as μ_i, while the degree of a meaning j is given as ω_j.

Figure 1.3 displays an example of a simple bipartite graph. A graphical representation is included, as well as the corresponding values of the mathematical concepts discussed in this section up to here.

1.1.2 Information theory

Information theory measures are used to compute the cost function of the optimization process. Here we go through a short overview of information theory formulas. They can be found in [6].

The general formula for the entropy of a set X is

$$H(X) = \sum_{x \in X} p(x) \log p(x) \tag{1.3}$$

where $p(x)$ is the probability associated with the element x. The general formula for the joint entropy of two sets X and Y is defined as

$$H(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{1.4}$$

where $p(x, y)$ is the joint probability of the elements x and y.

Any other information theoretic measures concerning two sets can be obtained from $H(X)$, $H(Y)$ and $H(X, Y)$. Mutual information

$$I(X, Y) = H(X) + H(Y) - H(X, Y),$$

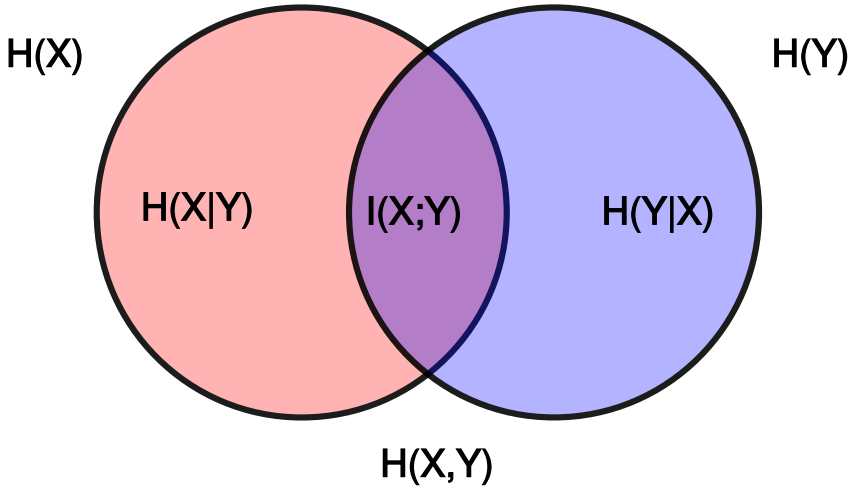


Figure 1.4: This diagram illustrates the relationships between information theoretic measures for two sets X and Y. The area occupied by both left and right circle is the joint entropy $H(X, Y)$. The circle on the left (both the red and the violet areas) represents the marginal entropy $H(S)$ while the circle on the right (both blue and violet areas) represents the marginal entropy $H(Y)$. The red area is the conditional entropy $H(X|Y)$ while the blue area represents the conditional entropy $H(Y|X)$. The violet area represents the mutual information $I(X, Y)$.

and conditional entropies

$$H(X|Y) = H(X, Y) - H(X)$$

and

$$H(Y|X) = H(X, Y) - H(Y).$$

The diagram on Figure 1.4 shows these relationships in a more visual way.

Based on Equation (1.3), we define the entropies of words and meanings, $H(S)$ and $H(R)$ respectively, as

$$H(S) = \sum_{i=1}^{n} p(s_i) \log p(s_i) \tag{1.5}$$

and

$$H(R) = \sum_{j=1}^{m} p(r_j) \log p(r_j). \tag{1.6}$$

We also define the joint entropy of words and meanings, from Equation (1.4), as

$$H(S, R) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(s_i, r_j) \log p(s_i, r_j). \tag{1.7}$$

1.1.3 The ϕ parameter

The ϕ parameter, which has already appeared up to this point, is used to generalize the older versions of the two main models studied.

In [10] and [13] these two models were introduced. Later, the ϕ parameter is added to generalize the model and hopefully be able to predict more linguistic laws. [14]

As is presented in the following sections, when $\phi = 0$, the older models are retrieved from the newer ones.

1.1.4 The internal model

In this model, the joint probability of a word s_i and a meaning r_j is proportional to the product of their degrees to the ϕ power. When $\phi = 0$, the joint probability depends only on whether there is a connection between the word and the meaning. Mathematically,

$$p(s_i, r_j) \propto a_{i,j} (\mu_i \omega_j)^{\phi}.$$

When $\phi = 0$ we recover a previous simpler model. [10]

The actual joint probability is derived from Equation (1.1.4), and from it the marginal probabilities of words and meanings are also derived. Ultimately this leads to the information theoretic equations seen in Section 1.1.2.

1.1.5 The external model

This model was introduced in [13] with meaning probabilities being constant and disconnected meanings being disallowed. Here we present a generalization of the previous model. In this model, the probability of a meaning r_j is given *a priori* as $\pi(r_j)$. It is possible for a meaning to be disconnected from any words, in which case $p(r_j) = 0$. In [13] disconnected meanings were disallowed and $\pi(r_j) = \frac{1}{m}$. Additionally, $\phi = 0$ in Equation (1.9). Here $\pi(r_j)$ can follow any probability distribution and disconnected meanings are allowed. Zero values of $\pi(r_j)$ are disallowed, however, as meanings that could never occur are not the target of communication.

We then define $p(r_j)$ as

$$p(r_j) \propto (1 - \delta_{\omega_j, 0}) \pi(r_j) \tag{1.8}$$

where $\delta_{a,b}$ is the Kronecker delta,

$$\delta_{a,b} = \begin{cases} 0 & \text{if } a \neq b \\ 1 & \text{if } a = b. \end{cases}$$

The word probability is then defined as the conditional probability of choosing a word given that a meaning has been chosen,

$$p(s_i|r_j) \propto a_{i,j}\mu_i^{\phi}. \tag{1.9}$$

With the marginal meaning probability and the conditional probability of a word given a meaning, the joint probability and marginal word probability can be calculated. From this the information theoretic equations are be obtained.

1.1.6 Optimization

As seen in previous sections and will be seen in Section 1.3, our hypothesis is that the empirical laws observed in human language are the result of minimizing the effort of speaker and hearer.

This effort is defined in terms of information theoretic measures. We choose two competing forces. $H(S)$, the entropy of the words, and $I(S,R)$, the mutual information between words and meanings. It is the goal of any communications system to maximize $I(S,R)$. The smaller $I(S,R)$, the greater the effort for the hearer. The higher the entropy of words, however, they harder they are to access. When $H(S) = 0$, only a single word has probability 1 while all others are 0, meaning that knowing which word to choose is trivial, while $H(S)$ is maximum when all words have equal probability, giving no indication of which should be chosen.

We define $\Omega(\lambda)$ as the cost function we aim to minimize,

$$\Omega(\lambda) = -\lambda I(S,R) + (1-\lambda)H(S) \tag{1.10}$$

where $0 \leq \lambda \leq 1$ is used to indicate the weight given to each of the two forces. When $\lambda = 0$, the minimization of $H(S)$ is completely favored while when $\lambda = 1$ the maximization of $I(S,R)$ is. When $\lambda = 1/2$, they are equally favored.

The optimization process follows a Markov Chain Monte Carlo method at zero temperature. It consists of making mutations to A, chancing $a_{i,j}$ from 1 to 0 or from 0 to 1. If a set of mutations results in a decrease of Ω, they are kept. Otherwise they are undone and another set is attempted.

1.1.7 Dynamic and static equations

As is seen later on, in Chapter 2, two sets of equations are derived for the information theoretic equations necessary to calculate Ω (Equation (1.10)).

A set of static equations recalculate everything from scratch. The set of dynamic equations calculate the changes done only after a mutation to A takes place.

	Internal model	External model
Static	$\mathcal{O}(M)$	$\mathcal{O}(M)$
Dynamic ($\phi \neq 0$)	$\mathcal{O}(\max(\mu_i, \omega_j))$	$\mathcal{O}(\max(\mu_i, \lvert B_{i,j} \rvert))$
Dynamic ($\phi = 0$)	$\mathcal{O}(1)$	$\mathcal{O}(\mu_i)$

Table 1.2: Summary table of the computational cost of each model and for particular cases. The columns indicate which of the two models and the rows the specific case for that model (static, dynamic general case for any value of ϕ and dynamic for the specific case of $\phi = 0$ where simplifications are possible). The internal model with $\phi = 0$ originates from [10]. The external model with $\phi = 0$ from [13]. The generalization where $\phi \neq 0$ for the family of models was introduced in [14] but without any simulation results. Each cell shows the computational complexity of updating the value of the cost function after mutating a_{ij} (that is, the word s_i and the meaning r_j become connected from being disconnected or become disconnected from being connected). M is the number of connections in the model. μ_i is the number of neighbors of the word (s_i) and ω_j the number of neighbors of the meaning (r_j). $|B_{i,j}|$ is the number of words that have at least one neighbor in common with s_i including r_j.

As seen previously this dynamic recalculation can be more efficient. In the case of this thesis, small changes are continuously done to the graph, making several mutations to A then reevaluating Ω. Dynamically accounting for the differences brought about for these changes is much more efficient than recalculating every single measure each time.

Dynamic calculation, however, brings its own set of problems, such as the increase in mathematical complexity and the introduction of greater floating point error. Section 1.2 goes over this in more detail while Section 3.1 covers the measures taken in order to deal with these problems.

Table 1.2 shows the computational costs of the two models for specific cases that are implemented in the C++ open source program. The justification can be found in Section 2.2.

1.2 Goals

Here the goals of the thesis are explained.

Older models [13] [10] where $\phi = 0$ can already make predictions. Can these more generic models predict anything new? And can they still make the predictions that the older models could? These older less general models already predicted Zipf's word frequency law. It would seem that the newer models should be able to make new predictions, such as the law of meaning frequency [14]. And perhaps other laws, such as the age frequency law. And of course, can they still predict the word frequency law? So one of the goals is to investigate the linguistic laws predicted by the model, to see if the more general version can still make the same prediction as the previous models. And also to

investigate whether it can make new predictions about other linguistic laws.

The computational aspect of the model should also be taken into account. The optimization process is key to the model, and it needs a stop condition. However, it is not obvious to know what the right stop condition is. The process might not reach a minimum due to not making enough attempts to continue minimizing the cost function before stopping. And due to the size of the search space it is unfeasible to exhaustively search every possible neighboring state. While the previous models with $\phi = 0$ were analyzed mathematically [19] [17], when $\phi \neq 0$ they become more complex. Another goal is to investigate whether the model reaches the local minimum.

Perhaps the main computational challenge is the recalculation of the cost function. It must be as efficient as possible, but it must also be precise to not compromise the results. To try to reduce the computational complexity of each step of the optimization process while keeping the results precise and relatively free of numerical error. This implies not only the derivation of all the mathematics necessary to calculate the cost functions of the models, but also additional mathematical work to derive dynamic equations allowing for only a small part of the cost function to be recalculated.

However, dynamic equations introduce more numerical error from the fact that they imply subtracting "old" values and adding "new" ones. Addition and subtraction operations can easily add numerical error to floating point operations. If not dealt with, this error can accumulate and change the results. An important goal, then, is to minimize numerical error originated by floating point operations without compromising efficiency.

Verification is also a very important point. The mathematics of the model are complex, specially the dynamic version. A single mistake can invalidate all the results obtained. This is why the implementation must be tested thoroughly.

There is an engineering component. The implementation and documentation of an open source tool allowing for the replication of the obtained results and also to further study these models.

This ties into another of the goals. Science must be reproducible. Previous work [13] [10] on this model suffered from a lack of reproducibility. The results of this thesis should be completely open and easily reproducible.

1.3 Hypothesis

Several hypothesis are stated here.

As seen in Chapter 1 and in [14], there is a relationship between the exponents of the Zipf's word-frequency and meaning-frequency laws. See Equation (1.2).

The main hypothesis of the thesis is that linguistic laws appear due to an optimization process in the association of words and meanings. In the model any effects of social interaction are neglected, which are the basis of other approaches to investigating human language such as the naming game. [2]

One of the questions that we seek to answer is whether local minima exist, as argued in [9].

It is also hypothesized that when $\phi = 1$ the models should be able to reproduce Zipf's meaning-frequency law, as seen in [14].

1.4 Outline of the thesis

The rest of the thesis is organized as follows:

In Chapter 2 the mathematical and computational aspects of the models are covered in detail. From the basic assumptions outlined in Section 1.1 the formulas for marginal and joint probabilities of words and meanings are derived and, from them, the information theoretic measures needed to compute the cost function. The dynamic equations are then derived from the static ones. The extreme cases (when there is only one connection between a word and a meaning or when every word is connected to every meaning, etc) are given as well as the invariants, the maximum and minimum possible values that the information theoretic measures can achieve. These steps are repeated for both the internal and external model. Along the way several properties are given, which are elaborated on in the Appendix along with some derivations that might be excessively long. For the computational aspect, the computational complexity of the static and dynamic versions of the computation of the cost function are shown. For the dynamic version a distinction is made between the general case and the case when $\phi = 0$. Again, the complexity is shown for both the internal and external model. Table 1.2 summarizes the complexity for the various methods of calculating the cost function in both models.

Chapter 3 goes into the more tangible details of the implementation of the model beyond just mathematics: the implementation decisions and various algorithms, a diagram of the class hierarchy of the implementation (Figure 3.1), the algorithms for the generation of the initial random graphs, the strategies used to deal with numerical error from floating point arithmetic are all outlined and the methods used to obtain several probability distributions for ϕ are all shown. The optimization algorithm is covered in more detail including the choices for a stop condition. The choices regarding parallelization of the algorithms and why it was decided to not make the algorithms parallel. Details about the verification of the model are given, including all the strategies used to ensure that the dynamic and static implementations were equivalent. The chapter closes with the problems encountered and how they were dealt with, both the numerical precision problems and the rest of problems encountered.

Chapter 4 presents the results obtained. Results from previously published papers [10] [13] are replicated. This both deals with the reproducibility issues in those works and serves as a benchmark to verify that the model is consistent with previous results. New data obtained for $\phi = 1$ and both models is presented. This data includes the evolution of the values of the information theoretic measures of the optimal graphs for values of λ ranging from 0 to 1. Certain statistical measures are shown for select values of λ, including several of

the relationships between frequency, frequency rank, degree (number of meanings of a word) and degree rank. Values of the exponents and factors of the observed power laws are obtained with a Theil-Sen estimator.

Chapter 5 discusses the results obtained and introduces some possible future work. Quantitative linguistics subjects are discussed, previously introduced word laws are compared with the data and it is argued whether the presented models can reproduce them and to what degree. These laws include the word frequency law, the meaning frequency and meaning distribution laws and the age frequency law. The computational results are also discussed, specifically regarding local minima of the cost function. The section closes with a review of future work. Alternative optimization methods are commented on, simulated annealing and gradient descent. Other possible predictions of the external model, such as the vocabulary learning already studied for the internal model [9] [4]. Possible further improvements to the numerical error problems are also commented.

Chapter 2

Model

TODO: reasoning of global optimum that is used to stop early

This chapter covers the details about the two models previously introduced in Chapter 1, where a high level introduction to both the internal model and the external model was given. It follows a top down approach. General concepts from both types of model are outlined first and the specific details from each model are given afterwards.

As a reminder of the previous chapter a very brief definition of the two models follows. Both models are quite similar, they represent a way to connect words with meanings. Both meanings and words are given probabilities of being used, from which information theoretic measures are derived, which are ultimately used to obtain a cost function. The high level differentiating trait is the way in which the probability of a meaning is obtained. In the first studied model (internal model), the probabilities of both words and meanings are *internal* to the model, they only depend on the structure of the bipartite graph. In the second model (external model), the probabilities of meanings are *external* to the model, they come from a distribution of *a priori* probabilities.

The remainder of this chapter is divided into two sections. Section 2.1 deals with the mathematical definitions of the models and it is more theoretical in nature. Section 2.2 outlines the computational aspects of the two models and it is more practical and implementation oriented. This chapter does not deal with any actual details of the implementation of the model. These details are given in Chapter 3.

Section 1.1 from Chapter 1 introduces the the graph theory and information theory concepts that will appear during the rest of this chapter.

2.1 Mathematical aspect

This section covers the mathematical definition of the model. Section 2.1.1 introduces concepts common to both models. These concepts are then extended for either the internal model or the external model in Sections 2.1.2 and 2.1.3

respectively.

2.1.1 Common concepts

Here a few more common concepts that were not discussed in Chapter 1 are introduced.

As a reminder, the notation defined for the degree of the vertices in the graph is repeated. The word s_i has degree μ_i, while the meaning r_j has degree ω_j. μ and ω are defined as

$$\mu_i = \sum_{j=1}^{m} a_{i,j} \tag{2.1}$$

and

$$\omega_j = \sum_{i=1}^{n} a_{i,j}. \tag{2.2}$$

There is an additional parameter of the models, ϕ, which is not directly related to the bipartite graph. This parameter appears in both models and it is used to "emphasize" the effect of the vertex degrees in the calculations. As explained in previous sections, this parameter is a new addition to the models originally presented in [10] and [13].

With the parameter ϕ, the definitions of μ and ω can be generalized. μ_ϕ and ω_ϕ are defined as

$$\mu_{\phi,i} = \sum_{j=1}^{m} a_{i,j}\omega_j^\phi \tag{2.3}$$

and

$$\omega_{\phi,i} = \sum_{i=1}^{n} a_{i,j}\mu_i^\phi. \tag{2.4}$$

It can be seen that when $\phi = 0$ Equation (2.3) becomes Equation (2.1) and Equation (2.4) becomes Equation (2.2). In other words, $\mu_{0,i} = \mu_i$ and $\omega_{0,j} = \omega_j$

2.1.2 The internal model

Here the concepts outlined in Section 1.1.4 are worked out into actual probabilities and information theoretic equations.

This model is defined by the joint probability of a word s_i and a meaning r_j seen in Equation (1.1.4). As seen in Section 1.1.4, this probability must be proportional to the product of the degrees of s_i and r_j.

From this definition equations for the marginal probabilities are derived, which in turn are used to obtain the information theory expressions in the cost function Ω (Equation (1.10)). From these expressions, dynamic equations are derived to obtain the change experienced by the entropies after a single mutation has occurred on the adjacency matrix A.

Joint Probability

Adding a normalizing factor M_ϕ, the joint probability becomes

$$p(s_i, r_j) = \frac{1}{M_\phi} a_{i,j} (\mu_i \omega_j)^\phi. \tag{2.5}$$

This normalizing factor is obtained by applying the definition of probability

$$\sum_{i=1}^{n} \sum_{j=1}^{m} p(s_i, r_j) = 1.$$

Then,

$$M_\phi = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{i,j} (\mu_i \omega_j)^\phi$$

or equivalently

$$M_\phi = \sum_{(i,j) \in E}^{n} (\mu_i \omega_j)^\phi. \tag{2.6}$$

Notably, when $\phi = 0$, M_ϕ is simply M and as follows from Equation (2.6), it's the total number of edges in the graph.

Marginal Probabilities

Formulas for the marginal probabilities $p(s_i)$ and $p(r_j)$ are derived from the joint probability using the general formula

$$p(x) = \sum_{y \in Y} p(x, y). \tag{2.7}$$

Applying Equation (2.7),

$$p(s_i) = \frac{\mu_i^\phi}{M_\phi} \sum_{j=1}^{m} a_{i,j} \omega_j^\phi.$$

Recall Equation (2.3),

$$p(s_i) = \frac{\mu_i^\phi \mu_{\phi,i}}{M_\phi}. \tag{2.8}$$

Symmetrically applying Equation (2.7) to obtain $p(r_j)$ and applying Equation (2.4) instead we obtain

$$p(r_j) = \frac{\omega_j^\phi \omega_{\phi,j}}{M_\phi}. \tag{2.9}$$

Entropies

$H(S,R)$ is obtained by applying the joint probability given in Equation (2.5) to Equation (1.7),

$$H(S,R) = -\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{M_\phi}a_{i,j}(\mu_i\omega_j)^\phi \log\left(a_{i,j}(\mu_i\omega_j)^\phi\right). \tag{2.10}$$

Property 1

This expression can be refined taking advantage of the equality

$$-\sum_i \frac{x_i}{T}\log\frac{x_i}{T} = \log T - \frac{1}{T}\sum_i x_i \log x_i.$$

This equality holds as long as

$$T = \sum_i x_i.$$

While is quite simple to prove that this is true, the derivation is given in Appendix B.1.1.

Applying Property 1 to Equation (2.10) we obtain

$$H(S,R) = \log M_\phi - \frac{\phi}{M_\phi}\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}\left((\mu_i\omega_j)^\phi \log a_{i,j}\mu_i\omega_j\right)$$

using E in the summation and adopting the convention that $0\log 0 = 0$

$$H(S,R) = \log M_\phi - \frac{\phi}{M_\phi}\sum_{(i,j)\in E}(\mu_i\omega_j)^\phi \log \mu_i\omega_j \tag{2.11}$$

is reached.

The marginal word entropy is found applying the marginal probability in Equation (2.8) to the definition of $H(S)$ in Equation (1.5)

$$H(S) = -\sum_{i=1}^{n}\frac{\mu_i\mu_{\phi,i}}{M_\phi}\log\frac{\mu_i\mu_{\phi,i}}{M_\phi}.$$

Property 1 can be applied immediately, obtaining

$$H(S) = \log M_\phi - \frac{1}{M_\phi}\sum_{i=1}^{n}\mu_i^\phi\mu_{\phi,i}\log\mu_i^\phi\mu_{\phi,i}. \tag{2.12}$$

The convention $0\log 0 = 0$ is applied here for disconnected words (where $\mu_i = 0$).

The marginal meaning entropy is found symmetrically, the marginal probability (Equation (2.9)) is applied to the definition of $H(R)$ (Equation (1.6)). With Property (1), we obtain

$$H(R) = \log M_\phi - \frac{1}{M_\phi} \sum_{j=1}^{m} \omega_j^\phi \omega_{\phi,j} \log \omega_j^\phi \omega_{\phi,j}. \tag{2.13}$$

also using the convention $0 \log 0 = 0$ for disconnected meanings ($\omega_j = 0$).

Dynamic Equations

The full derivation of the dynamic equations is given in [4]. Here only the final expressions of the dynamic equations are given.

Starting with compact expressions of the entropies (Equations (2.11), (2.12) and (2.13))

$$H(S,R) = \log M_\phi - \frac{\phi}{M_\phi} X(S,R) \tag{2.14}$$

$$H(S) = \log M_\phi - \frac{1}{M_\phi} X(S) \tag{2.15}$$

$$H(R) = \log M_\phi - \frac{1}{M_\phi} X(R) \tag{2.16}$$

with

$$X(S,R) = \sum_{(i,j)\in E} x(s_i, r_j) \tag{2.17}$$

$$X(S) = \sum_{i=1}^{n} x(s_i) \tag{2.18}$$

$$X(R) = \sum_{j=1}^{m} x(r_j) \tag{2.19}$$

$$x(s_i, r_j) = (\mu_i \omega_j)^\phi \log \mu_i \omega_j \tag{2.20}$$

$$x(s_i) = \mu_i^\phi \mu_{\phi,i} \log \mu_i^\phi \mu_{\phi,i} \tag{2.21}$$

$$x(r_j) = \omega_j^\phi \omega_{\phi,j} \log \omega_j^\phi \omega_{\phi,j}. \tag{2.22}$$

A prime mark is used to indicate a new value of a certain variable after a mutation has taken place. A variable without a prime mark indicates the value before the mutation took place. Suppose that $a_{i,j}$ mutates. Then

$$a'_{i,j} = 1 - a_{i,j} \tag{2.23}$$

$$\mu'_i = \mu_i + (-1)^{a_{i,j}} \tag{2.24}$$

$$\omega'_j = \omega_j + (-1)^{a_{i,j}} \tag{2.25}$$

We define the set of neighbors of any word s_i

$$\Gamma_S(i) = \{\, r_j \mid (s_i, r_j) \in E \,\}, \tag{2.26}$$

and similarly the set of neighbors of any meaning r_j

$$\Gamma_R(j) = \{\, s_i \mid (s_i, r_j) \in E \,\}. \tag{2.27}$$

Then, for any k such that $1 \leq k \leq n$, we have that

$$\mu'_{\phi,k} = \begin{cases} \mu_{\phi,k} - a_{ij}\omega_j^{\phi} + (1 - a_{ij}){\omega'_j}^{\phi} & \text{if } k = i \\ \mu_{\phi,k} - \omega_j^{\phi} + {\omega'_j}^{\phi} & \text{if } k \in \Gamma_R(j) \text{ and } k \neq i \\ \mu_{\phi,k} & \text{otherwise.} \end{cases} \tag{2.28}$$

Likewise, for any l such that $1 \leq l \leq m$, we have that

$$\omega'_{\phi,l} = \begin{cases} \omega_{\phi,l} - a_{ij}\mu_i^{\phi} + (1 - a_{ij}){\mu'_i}^{\phi} & \text{if } l = j \\ \omega_{\phi,l} - \mu_i^{\phi} + {\mu'_i}^{\phi} & \text{if } l \in \Gamma_S(i) \text{ and } l \neq j\ . \\ \omega_{\phi,l} & \text{otherwise.} \end{cases} \tag{2.29}$$

These variables can be applied to Equations (2.20), (2.21) and (2.22) directly to obtain their values.

We define the set $E_{i,j}$ as the set of all edges connecting s_i and r_j with their neighbors. That is,

$$E_{i,j} = \{\, (i,l) \mid l \in \Gamma_S(i) \,\} \cup \{\, (k,j) \mid k \in \Gamma_R(j) \,\} \tag{2.30}$$

With these definitions, we can now obtain the expressions of $X'(S,R)$ from $X(S,R)$ and of M'_ϕ from M_ϕ.

$$\begin{aligned} M'_\phi = M_\phi &- \left[\sum_{(k,l)\in E_{i,j}} (\mu_k \omega_l)^{\phi} \right] - a_{ij}(\mu_i \omega_j)^{\phi} \\ &+ \left[\sum_{(k,l)\in E_{i,j}} (\mu'_k \omega'_l)^{\phi} \right] + (1 - a_{ij})(\mu'_i \omega'_j)^{\phi}. \end{aligned}$$

Similarly, the new value of $X(S,R)$ will be

$$\begin{aligned} X'(S,R) = X(S,R) &- \left[\sum_{(k,l)\in E_{i,j}} x(s_k, r_l) \right] - a_{ij} x(s_i, r_j) \\ &+ \left[\sum_{(k,l)\in E_{i,j}} x'(s_k, r_l) \right] + (1 - a_{ij}) x'(s_i, r_j). \end{aligned} \tag{2.31}$$

The expressions of $X'(S)$ from $X(S)$ and $X'(R)$ from $X(R)$ used to dynamically update Equations (2.31), (2.32) and (2.33) are

$$X'(S) = X(S) - \left[\sum_{k\in\Gamma_R(j)} x(s_k)\right] - a_{ij}x(s_i) + \left[\sum_{k\in\Gamma_R(j)} x'(s_k)\right] + (1-a_{ij})x'(s_i) \tag{2.32}$$

and

$$X'(R) = X(R) - \left[\sum_{l\in\Gamma_S(i)} x(r_l)\right] - a_{ij}x(r_j) + \left[\sum_{l\in\Gamma_S(i)} x'(r_l)\right] + (1-a_{ij})x'(r_j). \tag{2.33}$$

Again, the full derivation of these equations is given in [4] and it is not included here to avoid repeating the same ideas.

The values of $H(S)$, $H(R)$ and $H(S,R)$ can be obtained by applying Equations (2.31), (2.32) and (2.33) to the definitions in Equations (2.14), (2.15) and (2.16) respectively.

Extreme cases and invariants

For verification purposes (see Section 3.4), the values of the entropies along with their invariants are also given here.

Extreme cases These cases can be easily derived from Equations (2.11), (2.12) and (2.13) ($H(S,R)$, $H(S)$ and $H(R)$ respectively) by assuming the corresponding condition and applying elementary algebra.

- For a single edge, $H(S), H(R), H(S,R) = 0$.
- For a complete graph, $H(S) = \log n, H(R) = \log m, H(S,R) = \log nm$
- For a one-to-one mapping of signals into meanings with $n = m$, $H(S), H(R), H(S,R) = \log n$

Invariants These invariants follow from basic information theory. [6]

- $0 \leq H(S) \leq \log n$
- $0 \leq H(R) \leq \log m$
- $0 \leq H(S,R) \leq \log nm$

2.1.3 The external model

Here the equations and concepts form 1.1.5 are worked out into probabilities and information theoretic equations.

In this model, $p(r_j)$ depends in an *a priori* probability $\pi(r_j)$ while $p(s_i|r_j)$ is defined in terms of μ_i^ϕ.

From the definition of probability,

$$\sum_{j=1}^{m} p(r_j) = 1, \tag{2.34}$$

even when there are disconnected meanings whose probability should be zero.

From Equation (1.8) and 2.34,

$$p(r_j) = \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho} \tag{2.35}$$

where

$$\begin{aligned} \rho &= \sum_{j=1}^{m} (1 - \delta_{\omega_j,0})\pi(r_j) \\ &= 1 - \sum_{j=1}^{m} \delta_{\omega_j,0}\pi(r_j). \end{aligned} \tag{2.36}$$

The conditional probability of choosing a word given a meaning is proportional to the number of meanings associated with that word or zero if that meaning is not associated to the word.

Applying

$$\sum_{i=1}^{n} p(s_i|r_j) = 1$$

to Equation (1.9) and recalling Equation (2.4) we obtain

$$p(s_i|r_j) = \frac{a_{i,j}\mu_i^\phi}{\omega_{\phi,j}}. \tag{2.37}$$

The joint probability $p(s_i, r_j)$ is obtained by applying Equation (2.35) and Equation (2.37) to the definition

$$p(s_i, r_j) = p(s_i|r_j)p(r_j),$$

obtaining

$$p(s_i, r_j) = \frac{a_{i,j}(1 - \delta_{\omega_j,0})\mu_i^\phi \pi(r_j)}{\rho\omega_{\phi,j}}. \tag{2.38}$$

Marginal word probability

The probability of a word can be derived from Equation (2.38) and applying

$$p(s_i) = \sum_{j=1}^{m} p(s_i, r_j),$$

obtaining

$$p(s_i) = \frac{\mu_i^{\phi} \chi_i}{\rho} \tag{2.39}$$

with

$$\chi_i = \sum_{j=1}^{m} \frac{a_{i,j}(1 - \delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}}. \tag{2.40}$$

Entropies

Applying the definition of $H(R)$ (Equation (1.6)) to $p(s_j)$ (Equation (2.35)) we obtain

$$H(R) = -\sum_{j=1}^{m} \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho} \log \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho}.$$

Property 1 can be applied immediately to simplify $H(R)$

$$H(R) = \log \rho - \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{\omega_j,0})\pi(r_j) \log(1 - \delta_{\omega_j,0})\pi(r_j).$$

This is simplified further applying the convention $0 \log 0 = 0$

$$\begin{aligned} H(R) &= \log \rho - \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{w_j,0})\pi(r_j) \log \pi(r_j) \\ &= \log \rho + \frac{1}{\rho} \left(H_\pi(R) + \sum_{j=1}^{m} \delta_{w_j,0} \pi(r_j) \log \pi(r_j) \right) \end{aligned} \tag{2.41}$$

where $H_\pi(R)$ is the entropy of the *a priori* probabilities

$$H_\pi(R) = -\sum_{j=1}^{m} \pi(r_j) \log \pi(r_j)$$

$H(S, R)$ can be derived by applying Equation (2.38) to the information theory definition of $H(S, R)$ (Equation (1.7)). Alternatively, it can also be derived by using the equality

$$H(S, R) = H(S|R) + H(R).$$

This reduces the problem to finding $H(S|R)$ using

$$H(S|R) = \sum_{j=1}^{m} H(S|r_j) p(r_j)$$

and

$$H(S|r_j) = -\sum_{i=1}^{n} p(s_i|r_j) \log p(s_i|r_j).$$

Both approaches turn out to be quite cumbersome mathematically. For the sake of brevity they are not shown here and can be consulted in Appendix B.2. In either case, the resulting expression for $H(S,R)$ is

$$H(S,R) = \log\rho - \frac{1}{\rho}\sum_{j=1}^{m}(1-\delta_{w_j,0})\pi(r_j)\left[\frac{\phi\nu_j}{\omega_{\phi,j}} + \log\frac{\pi(r_j)}{\omega_{\phi,j}}\right] \tag{2.42}$$

with

$$\nu_j = \sum_{i=1}^{n} a_{i,j}\mu_i^{\phi}\log(\mu_i). \tag{2.43}$$

$H(S)$ is derived quite easily by applying Equation (2.39) to the information theory definition (Equation (1.5))

$$H(S) = -\sum_{i=1}^{n}\frac{\mu_i^{\phi}\chi_i}{\rho}\log\frac{\mu_i^{\phi}\chi_i}{\rho}$$

Property 2

It is quite simple to see that

$$\sum_{i=1}^{n}\mu_i^{\phi}\chi_i = \rho,$$

however this might not be immediately obvious as it has not appeared at any point until now. Appendix B.1.2 shows that this property indeeed holds.

Following Property 2, Property 1 can be applied here, obtaining

$$H(S) = \log\rho - \frac{1}{\rho}\sum_{i=1}^{n}\mu_i^{\phi}\chi_i\log\mu_i^{\phi}\chi_i \tag{2.44}$$

Dynamic Equations

The full derivation of the dynamic equations is given in Appendix B.3. Here only the final expressions of the dynamic equations are given. Recall Section 2.1.2 for many useful definitions. In all dynamic equations, a mutation on $a_{i,j}$ is assumed.

Compact expressions of the entropies (Equations (2.41), (2.42) and (2.44))

$$H(S,R) = \log\rho - \frac{1}{\rho}X(S,R) \tag{2.45}$$

$$H(S) = \log\rho - \frac{1}{\rho}X(S) \tag{2.46}$$

$$H(R) = \log\rho - \frac{1}{\rho}X(R) \tag{2.47}$$

with

$$X(S,R) = \sum_{j=1}^{m}(1-\delta_{\omega_j,0})x(r_j) \tag{2.48}$$

$$X(S) = \sum_{i=1}^{n} x_{s_i} \tag{2.49}$$

$$X(R) = \sum_{j=1}^{m}(1-\delta_{\omega_j,0})\pi(r_j)\log\pi(r_j) \tag{2.50}$$

$$x(s_i) = \mu_i^{\phi}\chi_i \log\left(\mu_i^{\phi}\chi_i\right) \tag{2.51}$$

$$x(r_j) = \pi(r_j)\left[\frac{\phi\nu_j}{\omega_{\phi,j}} + \log\frac{\pi(r_j)}{\omega_{\phi,j}}\right]. \tag{2.52}$$

Equations (2.23) ($a'_{i,j}$) and (2.24) (μ'_i) remain the same as in the internal model. Equation (2.25) (ω'_j) is unused in this model.

Equation (2.29) ($\omega'_{\phi,l}$ for any l such that $1 \leq l \leq m$) remains identical, while Equation (2.28) ($\mu'_{\phi,k}$ for any k such that $1 \leq k \leq n$) is unused in this model.

Recall as well the definition of the set of all edges to neighbors of s_i and r_j (Equation (2.30)). Two additional sets are defined for ease in the definition of these equations, $A_{i,j}(k)$ is the set of all meanings that are neighbors of i ($\Gamma_S(i)$) (plus the meaning r_j if not already included) that are also neighbors of k

$$A_{i,j}(k) = (\Gamma_S(i) \cup \{\, r_j \,\}) \cap \Gamma_S(k). \tag{2.53}$$

The other set, $B_{i,j}(k)$ is the set of all words s_k such that $A_{i,j}(k)$ is not the empty set. The word s_i is always included,

$$B_{i,j}(k) = \{\, s_k \mid A_{i,j}(k) \neq \emptyset \,\} \cup \{\, s_i \,\} \tag{2.54}$$

The one mutation change equations for ρ, ν and χ are

$$\rho' = \rho - \delta_{\omega'_j,0}\pi(r_j) + \delta_{\omega_j,0}\pi(r_j), \tag{2.55}$$

$$\nu'_l = \begin{cases} \nu_l + (1-a_{ij})\mu'^{\phi}_i \log\mu'_i - a_{ij}\mu_i^{\phi}\log\mu_i & \text{if } l = j \\ \nu_l - \mu_i^{\phi}\log\mu_i + \mu'^{\phi}_i\log\mu'_i & \text{if } l \in \Gamma_S(i) \text{ and } l \neq j \\ \nu_l & \text{otherwise} \end{cases} \tag{2.56}$$

and

$$\chi'_k = \chi_k - \sum_{l\in A_{i,j}(k)} \frac{\pi(r_l)}{\omega_{\phi,l}} + \sum_{l\in A_{i,j}(k)} \frac{\pi(r_l)}{\omega'_{\phi,l}}. \tag{2.57}$$

χ_i is always recalculated statically instead

These variables can be applied to Equations (2.51) and (2.52) directly to obtain their values.

The expressions of $X'(S)$, $X'(R)$ and $X'(S,R)$ used to dynamically update Equations (2.41), (2.42) and (2.44) are

$$X'(R) = X(R) - \delta_{\omega'_j,0}\pi(r_j)\log\pi(r_j) + \delta_{\omega_j,0}\pi(r_j)\log\pi(r_j), \tag{2.58}$$

$$X'(S,R) = X(S,R) - \sum_{l\in\Gamma_S(i)\setminus\{j\}} x(r_l) + \sum_{l\in\Gamma_S(i)\setminus\{j\}} x'(r_l) - (1-\delta_{\omega_j,0})x(r_j) + (1-\delta_{\omega'_j,0})x'(r_j) \tag{2.59}$$

and

$$X'(S) = X(S) - \sum_{o\in B_{i,j}(k)} x(r_o) + \sum_{o\in B_{i,j}(k)} x'(r_k) \tag{2.60}$$

Again, the full derivation and logic behind these equations is given in Appendix B.3. It is not included here for brevity, as it is a long not immediately obvious derivation.

Extreme cases and invariants

For verification purposes (see Section 3.4), the values of the entropies along with their invariants are also given here.

Extreme cases These cases can be easily derived from Equations (2.42), (2.44) and (2.41) ($H(S,R)$, $H(S)$ and $H(R)$ respectively) by assuming the corresponding condition and applying elementary algebra.

- For a single edge, $H(S), H(R), H(S,R) = 0$.
- For a complete graph, $H(S) = \log n, H(R) = H_\pi(R), H(S,R) = H_\pi(R) + \log n$
- For a one-to-one mapping of signals into meanings with $n = m$, $H(S), H(R), H(S,R) = H_\pi(R)$

Invariants These invariants follow from basic information theory. [6]

- $0 \leq H(S) \leq \log n$
- $0 \leq H(R) \leq H_\pi(R)$
- $0 \leq H(S,R) \leq H_\pi(R) + \log n$

2.1.4 Lower bound of the cost function

Upper and lower bounds for the cost function Ω can be derived from the definition of Omega (Equation (1.10)) and the bound [6]

$$0 \leq I(S,R) \leq \min(H(S), H(R)),$$

obtaining

$$\begin{aligned}\Omega(\lambda) &\geq -\lambda \min(H(S), H(R)) + (1-\lambda)H(S) \\ &\geq (1-2\lambda)H(S)\end{aligned}$$

Knowing the bounds for $H(S)$ (Section 2.1.3) we reach that, in general

$$\Omega(\lambda) \geq \begin{cases} 0 & \text{if } \lambda \leq 1/2 \\ (1-2\lambda)\log\min(n,m) & \text{if } \lambda \geq 1/2. \end{cases} \tag{2.61}$$

2.2 Computational aspect

This section covers the computational side of the model. This side is based on the mathematics covered in Section 2.1. Both models are seen separately, Section 2.2.1 covers the internal model while Section 2.2.2 covers the external model. Both sections cover the computational cost of computing the variables of the model, both completely (static calculation) and the change after a mutation to $a_{i,j}$ (dynamic calculation) as well as the changes that take place by treating the case $\phi = 0$ separately.

A summary table comparing the computational costs of each of the models and the particular cases here is given at the end as Table 1.2.

2.2.1 The internal model

This section covers the computational cost of the calculation of the entropies of the internal model.

Static

A, E, μ and ω are always calculated dynamically and updated whenever an edge is added or removed to the graph, as it is very simple to do so. All entropies are calculated statically in a single loop iterating over every edge in E. The joint entropy $H(S,R)$ (Equation (2.11)) and the normalization factor M_ϕ (Equation (2.6)) are updated on every step of the loop. μ_ϕ and ω_ϕ are updated on every iteration as well. $H(S)$ (Equation (2.12)) is only updated when $\mu_{\phi,i}$ for a particular word s_i has been fully recalculated. $H(R)$ (Equation (2.13)) is updated in the same way as $H(S)$, whenever $\omega_{\phi,j}$ for a particular meaning r_j has been fully recalculated. The cost of the static calculation of entropies is $\mathcal{O}(M)$.

Dynamic

In the case of the dynamic calculation of entropies, the algorithmic cost is dominated by the computation of $X'(S)$, $X'(R)$ and $X'(S,R)$ (Equations (2.31), (2.32) and (2.33)). It can be seen immediately that looping over all the neighbors of s_i and r_j is necessary in order to update all entropies. The cost of the dynamic calculation of entropies is then $\mathcal{O}(\max(\mu_i, \omega_j))$.

The case $\phi = 0$

In order to speed up calculations, simpler equations for the case $\phi = 0$ are derived. $X'(S, R)$ (Equation (2.31)) is no longer used, as $H(S, R) = \log M$ when $\phi = 0$ (see Equation (2.11)).

$X'(S)$ (Equation (2.32)) becomes

$$X'(S) = X(S) - \mu_i \log \mu_i + \mu'_i \log \mu'_i \tag{2.62}$$

using the convention $0 \log 0 = 0$ where necessary.

Similarly to $X'(S)$, $X'(R)$ (Equation (2.33)) becomes

$$X'(R) = X(R) - \omega_j \log \omega_j + \omega'_j \log \omega'_j \tag{2.63}$$

In the case $\phi = 0$, M_ϕ (Equation (2.6)) becomes M, the number of edges in the graph.

As can be seen from equations (2.62) and (2.63), the cost of the dynamic calculation is greatly reduced when $\phi = 0$, becoming $\mathcal{O}(1)$.

2.2.2 The external model

This section overs the computational cost of the calculation of the entropies of the external model.

Static

As with the internal model (see Section 2.2.1), A, E, μ and ω are dynamically updated whenever an edge is added or removed from the graph while all other variables, including entropies, are calculated statically.

As can be seen from Equations (2.41), (2.42) and (2.44), a loop over every edge in the graph is needed in order to recalculate all entropies and variables. Unlike the internal model, however, this loop needs to be repeated twice.

During the first run, ω_ϕ (Equation (2.4)) and ν (Equation (2.43)) are updated on every iteration. ρ (Equation (2.36)), $H(R)$ (Equation (2.41)) and $H(S, R)$ (Equation (2.42)) are calculated only on the iterations where $\omega_{\phi,j}$ and ν_j have been completely calculated for a single meaning r_j.

This leaves χ and $H(S)$ to be calculated during the second run of the loop over all edges. The computation of χ_i (Equation (2.40)) for a word s_i requires $\omega_{\phi,j}$ to be fully computed for every meaning r_j. This is the reason for running two separate loops, fully calculating ω_ϕ in one run so that χ may be calculated in the other. $H(S)$ (Equation (2.44)) is updated only on the iterations where χ_i has been completely calculated for a single word s_i.

The cost of the static calculation of entropies is the same as in the internal model, $\mathcal{O}(M)$.

Dynamic

The dynamic calculation of entropies is dominated by the computation of $X'(S)$, $X'(S,R)$ and χ (Equations (2.48), (2.49) and (2.57)). These three values iterate on three different sets: $\Gamma_S(i)$, $A_{i,j}$ and $B_{i,j}$ (Equations (2.26), (2.53) and (2.54)). It is clear from the definition of set $A_{i,j}$ is lesser or similar (one more element) in size to the set $\Gamma_S(i)$, but $\Gamma_S(i)$ will usually be bigger. The size of the set $B_{i,j}$ depends on the structure of the graph.

The cost of the dynamic calculation of entropies is then $\mathcal{O}(\max(\mu_i, |B_{i,j}|))$

The case $\phi = 0$

In order to speed up calculations, simpler equations for the case $\phi = 0$ are derived. Much of the dependency on neighboring nodes is removed.

$X(R)$ and ρ (Equations (2.58) and (2.55)) are not affected as they do not depend on ϕ. They remain simple.

$x(r_j)$ becomes

$$(1 - \delta_{w_j,0})\pi(r_j) \log \frac{\pi(r_j)}{\omega_j} \tag{2.64}$$

and so no longer depends on neighbors of r_j. Consequently, $X(S,R)$ also no longer depends on a set of neighbors and becomes

$$X'(S,R) = X(S,R) - x(r_j) + (1 - \delta_{\omega'_j,0})x'(r_j). \tag{2.65}$$

χ_i (Equation (2.40)) is simplified but still depends on the neighborhood of s_i in order to be updated

$$\chi'_k = \begin{cases} \chi_i + (1 - a_{i,j})\frac{\pi(r_j)}{\omega'_j} - a_{i,j}\frac{\pi(r_j)}{\omega_j} & \text{if } k = 1 \\ \chi_k - \frac{\pi(r_j)}{\omega_j} + \frac{\pi(r_j)}{\omega'_j} & \text{if } k \in \Gamma_R(j) \text{ and } k \neq i \\ \chi_k & \text{otherwise.} \end{cases} \tag{2.66}$$

As $X(S)$ depends on χ (Equation (2.49)), $X'(S)$ becomes

$$X'(S) = X(S) - \sum_{k \in \Gamma_R(j) \cup \{ i \}} x(s_i) + \sum_{k \in \Gamma_R(j) \cup \{ i \}} x'(s_i) \tag{2.67}$$

with (see Equation (2.51))

$$x(s_i) = \chi_i \log \chi_i. \tag{2.68}$$

$X(S)$ and χ need to iterate the neighbors of s_i, and so the computational complexity is $\mathcal{O}(\mu_i)$.

Chapter 3

Methods

Pendent: parlar sobre theil sen ???

This chapter centers on the implementation of the models defined in Chapter 2 and the optimization method outlined in Section 1.1.6. Section 3.1 focuses on the implementation of the models in a C++ program, including a class hierarchy diagram, several important algorithms, various methods used to avoid floating point error and the different distributions of π (see Section 2.1.3) that were implemented. Section 3.2 deals with the optimization algorithm, how it changes for the static or dynamic cases and several optimizations that allow for faster runtime and more precise results. On Section 3.3 the approach taken to parallelization is outlined and justified. Section 3.4 deals with the verification of the code. Finally, Section 3.5 explains the main challenges related to numerical precision while Section 3.6 covers other problems related to the implementation.

3.1 Model implementation

This section covers the implementation of the two models seen in Chapter 2. It starts with a class hierarchy diagram of the organization of the static and dynamic versions of the two models. Following this, the algorithm used to randomly generate a random bipartite graph (given a vertex probability or given the number of edges) is outlined. Following this, the various "tricks" used in order to minimize the floating point error, specially in the dynamic version of the model are explained. To close this section, details on the various distributions of π that were implemented are given.

Class hierarchy diagram

The diagram in Figure 3.1 shows the classes implementing the bipartite graph models (see Chapter 2).

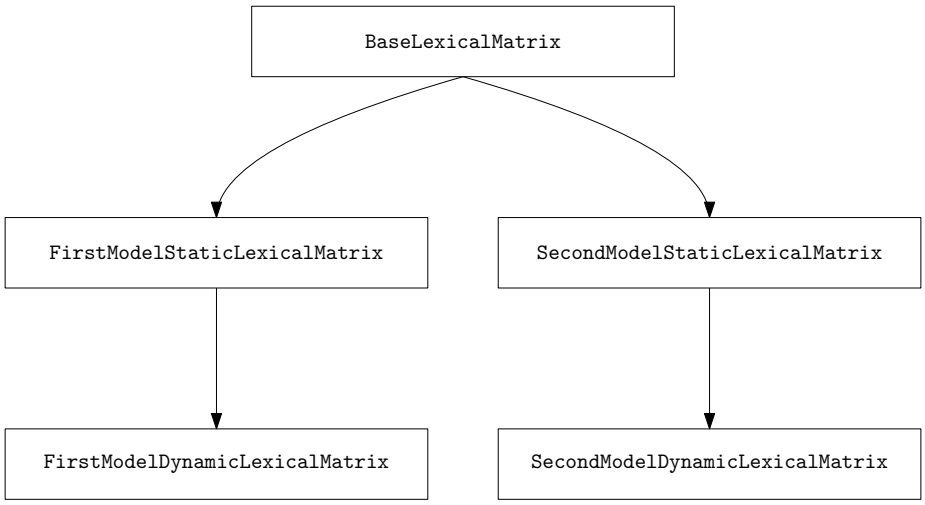


Figure 3.1: Diagram showing the hierarchy of the classes implementing the two models in their dynamic and static variants in the C++ program.

Random graph generation

This section outlines the algorithm implemented to generate a random Erdos-Renyi bipartite graph [7] in pseudocode. The graph can be generated by specifying either the probability of edge creation ($G_{n,m,p}$) as seen in Algorithm 1 or the total number of edges ($G_{n,m,e}$) as in Algorithm 2. Note that all Algorithm 1 actually does is compute a value of e then call Algorithm 2.

Algorithm 1 $G_{n,m,p}$ algorithm to generate a random $n \times m$ graph given probability of edge p.

1: **procedure** $G(n, m, p)$
2: $e \leftarrow \text{BINOM}(n \cdot m,\ p)$
3: **return** $\mathrm{G}(n, m, e)$
4: **end procedure**

Dealing with floating point error

One of the major challenges (see Section 3.5 has been dealing with floating point error. Two major "tricks" have been used in order to minimize this loss of precision as much as possible.

The Accumulator class Whenever possible, specially in the static calculations, additions have been done through an accumulator class. This class maintains two variables, `total` and `current`. When a value is added, it is added to

Algorithm 2 $G_{n,m,e}$ algorithm to generate a random $n \times m$ graph given the number of edges e.

```
1: procedure G(n, m, e)
2:     if disconnected meanings disallowed then
3:         for each r_j such that 1 ≤ j ≤ m do
4:             s_i ← UNIFORM(s_1, s_n)
5:             ADD EDGE(s_i, r_j)
6:             e ← e − 1
7:         end for
8:     end if
9:     while e > 0 do
10:        s_i ← UNIFORM(s_1, s_n)
11:        r_j ← UNIFORM(r_1, r_n)
12:        ADD EDGE(s_i, r_j)
13:        e ← e − 1
14:    end while
15: end procedure
```

the `current` variable. If the `current` variable is greater than `total`, it is added to `total` and reset to 0. This is an inexpensive and effective way to minimize the loss of precision from a common occurrence during static calculation, the addition of a smaller value into a greater one.

Static recalculation During the optimization process, whenever a new minimum is reached, a static recalculation is forced. In this way, one of the major source of floating point error from the dynamic version is removed. The dynamic equations (see Section 2.1) consist of many additions and subtractions. These additions and subtractions are a major contributor to the accumulation of floating point error. Since finding a new minimum is not common, this does not affect the run time of the optimization process.

Save and restore Most steps of the optimization process will not improve the cost function (see Section 3.2). Instead of undoing the step and recalculating all the parameters of the model, the parameters are saved before performing the step, and they are restored when the step needs to be reverted. This strategy reduces the amount of additions and subtractions to be done when the recalculation of parameters is dynamic, thus reducing the amount of floating point error.

Distribution of π

The second model (see Section 2.1.3) is based on a vector of length m of *a priori* probabilities π. Uniform, geometric, power law and broken stick probability distributions have been implemented in order to give values to the elements of this vector.

Uniform distribution The uniform distribution simply assigns the probability $1/m$ to every element in the vector.

Geometric distribution Assigns a right truncated geometric distribution with parameter p to the elements of π, following

$$(1-p)^{x-1}\frac{p}{1-(1-p)^m}.$$

This formula can be easily derived from the more general formula for a truncated discrete probability distribution

$$P(X = x|a < x \leq b) = p(X = x)/(P(X = b) - P(X = a)).$$

Power law distribution Assigns a right truncated power law distribution with parameter α to the elements of π. It follows

$$x^{-\alpha}\sum_{i=1}^{m} i^{-\alpha}$$

Broken stick distribution A broken stick distribution corresponds to the distribution of the sizes of the pieces of a stick that is broken by a point along its length randomly a number $m-1$ of times. This distribution corresponds to the expected value of the lengths of the pieces of a stick of total length 1. The formula and its derivation is given in [22] (Equation 1).

3.2 Optimization

The optimization algorithm follows the Markov Chain Monte Carlo method at zero temperature. The model is initialized in some way (depending on parameters), and on each iteration a number of random mutations (adding or removing an edge on the underlying bipartite graph) are effectuated. The cost function is evaluated with a parameter λ. If it is found to have improved, the model is recalculated statically (to avoid floating point error, see Section 3.1). If it is found to not have improved, the changes are reverted. The optimization stops after a number of failures to improve the cost function. The optimization stops early if the lower bound of the cost function is reached (See Equation (2.61)).

The parameter λ is given as a parameter. It controls the weight of the two forces that contribute to the cost function. See Section 1.1.6.

The number of mutations done on each step depends on the parameters and may be either constant or follow a binomial distribution. In the case of a binomial distribution, it is possible that zero mutations may be effectuated. In this case, nothing is done and the step is considered to not have improved the cost function.

The number of failures to improve the cost function needed to stop iterating depends on parameters. Two possible values are calculated from parameters of the model:

- **weak**: Enough steps are performed such that it is likely that all edges have been visited once. This follows the Coupon Collector's Problem [16], and the formula for the number of attempts is

$$\lfloor nm \log nm \rfloor$$

- **strong**: Enough steps are performed such that it is likely that every possible pair of edges has been visited once. This also follows the Coupon Collector's Problem,

$$\left\lfloor \binom{nm}{2} \log \binom{nm}{2} \right\rfloor$$

3.3 Parallelization

During the initial phases of the design of the software, it was considered to add parallelization in order to speed up calculations. Parallelization would add overhead and complexity to an already complex codebase in exchange for sharing the computation between the machine's processors.

A typical task for the program is the computation of a range of values of λ (see Section 3.2) and to calculate several samples for each λ. This layout lends itself well to multiprocessing, without the program implementing any sort of parallelization scheme.

The user is expected to divide the load his or herself, for instance by generating many separate configuration files (or using a script to do so) and using a tool such as one of the many implementations of `parallel` or a computing cluster's workload manager. In this way, the operating system handles the parallelization complexity and the software's complexity and overhead is reduced. This also makes verification much easier.

3.4 Verification

Verification is a vital part of any software development process. In this case, verification involves ensuring that both static and dynamic versions of the model are correctly implemented. Dynamic versions are specially complex and so need to be tested thoroughly.

The test routines implemented verify that the models give "sane" results by checking extreme cases (see Sections). In addition, the invariants of the entropies (also in Sections) are verified after every mutation and the program is aborted if they ever do not hold. A test also checks the invariants exhaustively for every possible combination of connected and disconnected edges for very small graphs ($n = m = 4$).

Other functionality less related with the implementation of the model is also tested.

- The save/restore functionality described in Section 3.1 for dealing with floating point error.

- Quickly testing whether the matrix has disconnected meanings (configuration parameter disallows disconnected meanings).

The last test is running the program for relatively small values of n and m and all combinations of parameters, and exhaustively checking that both static and dynamic versions give identical output for the same seed. This test is used to ensure the correctness of the dynamic version.

3.5 Numerical precision problems

Challenges and problems related to the numerical precision are given in this section.

In both the static and dynamic versions of the calculations, there are many additions or subtractions of floating point numbers. These actions often involve a loss of precision due to the difference in magnitude of the values being operated. Over many iterations, this loss of precision can become apparent and change the final result of the simulation. See Section 3.1 where an entire subsection is dedicated to dealing with numerical error.

The geometric distribution decreases in probability very quickly. Because of this, erroneous values can be produced when the value of π is too close to 0. As outlined in Section 2.1.3, the values of π should never be zero. This problem occurs in general whenever π is too close to zero and the program automatically warns about it if the parameters will generate values of π that are too small. For this reason, the geometric distribution does not appear in Chapter 4

3.6 Other problems

Since the runtime of the dynamic versions often depends on the number of neighbors of the affected vertices (see Section 2.2), running calculations on complete or almost complete graphs takes a long time, as the advantages of the dynamic formulas are reduced significantly. As a result of this, running the optimization with an initial complete graph can take an unacceptable amount of time when compared to other initial conditions.

The strong stop condition (Section 3.2) results far too many attempts to improve the cost function when $n = m$ is large (for $n = m = 400$, the number of attempts is of the order of 10^{12}). For this reason, the strong stop condition is not used in Chapter 4

Chapter 4

Results

In this chapter the results generated with the open source tool are presented. This chapter is divided into two sections. Section 4.1 presents results from previous papers recreated using the tool. Section 4.2 presents results from the newer family of models introduced in [14]

4.1 Verification of previous results

In this section a replication of previous results is presented. This allows to verify the model and its implementation against previous experiments.

However, in some cases there was limited information in the original papers about the parameters used in the experiments. This concerned details regarding the initial graphs and the specifics of the optimization process. In other cases, there were errors in previous papers. This was due to undetected programming errors. In both cases, determining the correct parameter becomes a matter of trial and error.

The ϕ parameter was not considered in either of the papers replicated in this section, and so it is set to 0.

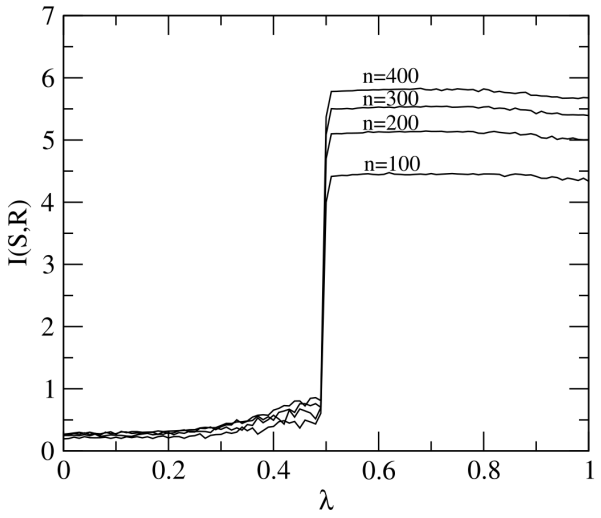
Section 4.1.1 replicates the results from [10], which correspond to the internal model from this thesis. Section 4.1.2 replicates the results from [13], corresponding to the external model.

4.1.1 Results from the internal model (2005)

Figure 4.1 shows both the original Figure 2 from [10] and a recreation of this figure using the new tool. Figure 4.2 shows both the original Figure 3 from [10] and a recreation of this figure, also generated using the new tool.

The initial graph was not specified in [10] and a $G_{n,m,1/n}$ $(n = m)$ graph is used in the replication. The paper also does not specify how to stop the optimization process and so the weak stop condition (Section 3.2) is used.

a)



b)

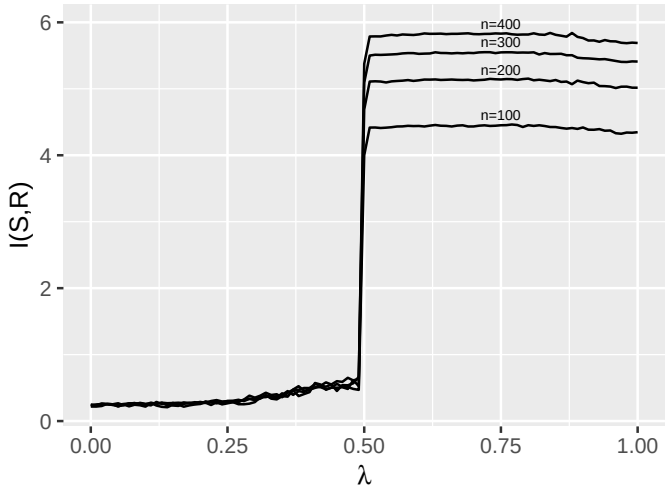
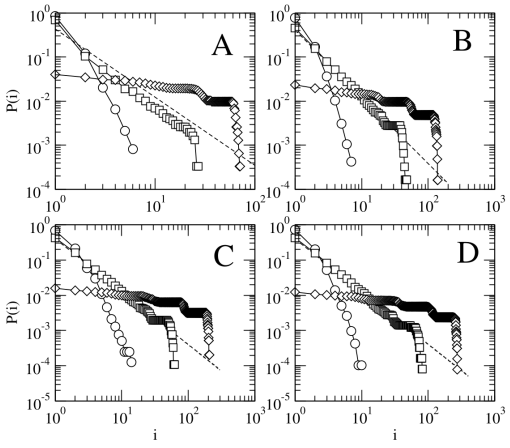


Figure 4.1: The mutual information $I(S, R)$ is in the y axis and the λ parameter on the x axis. Graphs of different sizes are shown, $n = m = 100$, $n = m = 200$, $n = m = 300$, $n = m = 400$. Averages over 30 realizations. $\phi = 0$, the initial graph is $G_{n,m,1/n}$, each iteration of the optimization algorithm performs 2 mutations on the graph, the weak stop condition is used to stop the optimization process, unlinked objects are allowed.
Subfigure a corresponds with Figure 2 from [10].
Subfigure b is the recreation of that same figure.

a)



b)

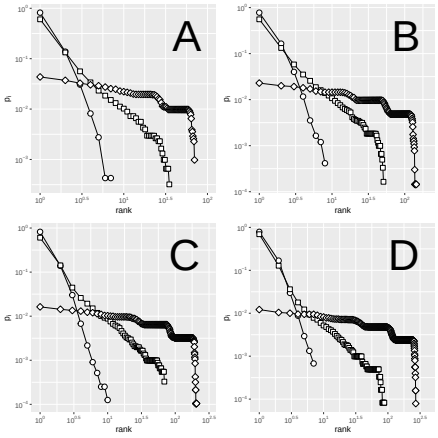


Figure 4.2: $P(i)$, the probability of the ith most frequent signal, obtained from minimum energy configurations for systems of sizes $n = m = 100$ (A), $n = m = 200$ (B), $n = m = 300$ (C), $n = m = 400$ (D). Four series are shown in each plot, $\lambda = 0.49$ (circles), $\lambda = \lambda^*$ (squares) and $\lambda = 0.5$ (diamonds) and the ideal curve for α^*. Averages over 30 realizations. When $\lambda = \lambda^*$, $\alpha^* = 1.54$ for $n = m = 100$, $\alpha^* = 1.51$ for $n = m = 200$, $\alpha^* = 1.5$ for $n = m = 300$ and $\alpha^* = 1.49$ for $n = m = 400$. It was chosen that $\lambda^* = 0.4986$ for $n = m = 100$, $\lambda^* = 0.4987$ for $n = m = 200$, $\lambda^* = 0.4987$ for $n = m = 300$, and $\lambda^* = 0.4986$ for $n = m = 400$. Averages over 30 realizations. $\phi = 0$, the initial graph is $G_{n,m,1/n}$, each iteration of the optimization algorithm performs 2 mutations on the graph, the weak stop condition is used to stop the optimization process, unlinked objects are allowed.
Subfigure a corresponds with Figure 3 from [10].
Subfigure b is the recreation of that same figure.

Subfigures a and b from Figure 4.1 are nearly identical. In Figure 4.2, subfigures a and b are also qualitatively very similar. Although some of the points can be seen to be not quite in the same place.

4.1.2 Results from the external model (2003)

Figure 4.3 shows both the original Figure 2 from [13] and the recreation of this figure using the new tool. Figure 4.4 shows the original Figure 3 from [13] and a recreation using the new tool.

The initial graph was specified as a $G_{n,m,\rho}$ but the value of ρ was not given in the paper, a $G_{n,m,10/n}$ graph was used (that is, a value of $\rho = \frac{10}{n}$). It is specified that the number of mutations done on each iteration of the optimization algorithm follows a binomial distribution. However it was later found that, due to a bug in the generator of binomial numbers, the number of mutations must have been lower. Several values were tried for the average number of binomial mutations, with 1.75 giving the results most similar to the previous model's.

Qualitatively, it seems that the two subfigures in Figure 4.3 are very similar. Some of the points in A do not form exactly the same shape, however, and it seems that not as many points appear inside the phase transition. As for Figure 4.4, subfigures A and C are quite similar (although the dashed line appears to overlap the data in subfigure b more than it did in a). Subfigure B is not as similar. However, the slope of the curve is also 1.0 as indicated by the solid line.

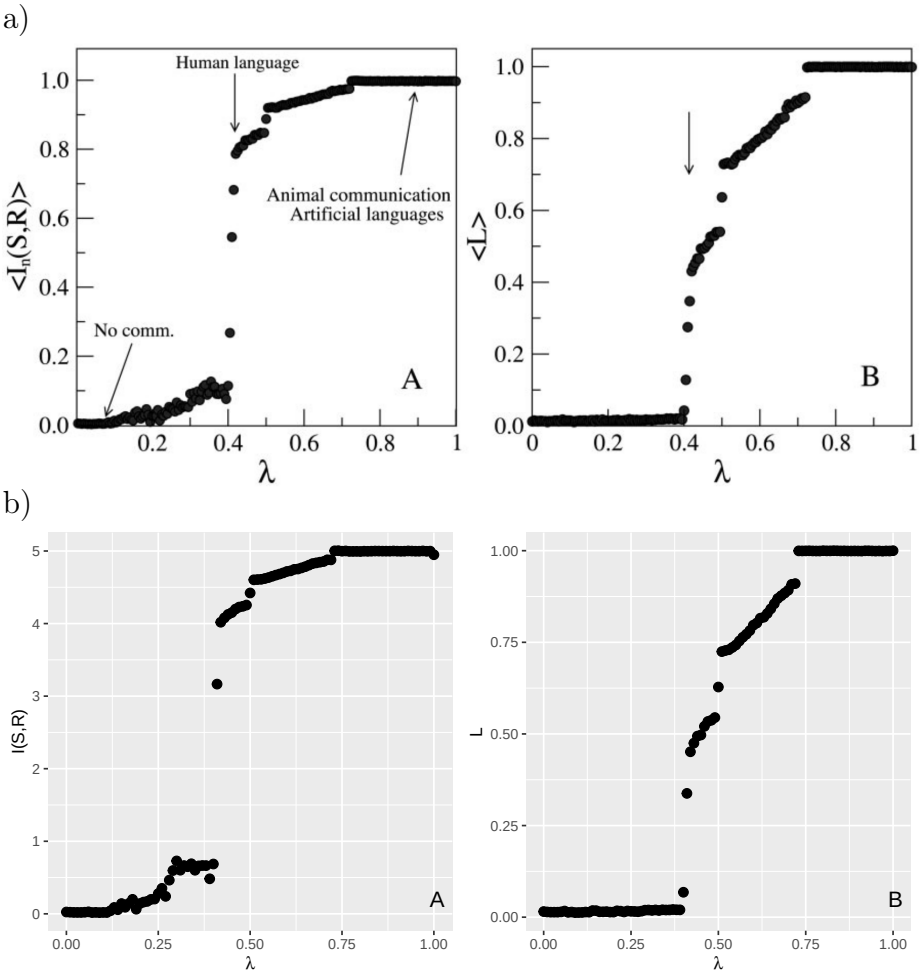


Figure 4.3: In A, $I(S,R)$ is the mutual information obtained for values of λ between 0 and 1. In B, L is the lexicon size obtained for values of λ between 0 and 1. An abrupt change is seen for $\lambda \approx 0.41$ in both A and B. Averages over 30 replicas. $n = m = 150$, $\phi = 0$, unlinked objects are not allowed, π follows a uniform distribution.
Subfigure a corresponds with Figure 2 from [13]
Subfigure b is the recreation of that same figure using the open source tool created for this thesis.

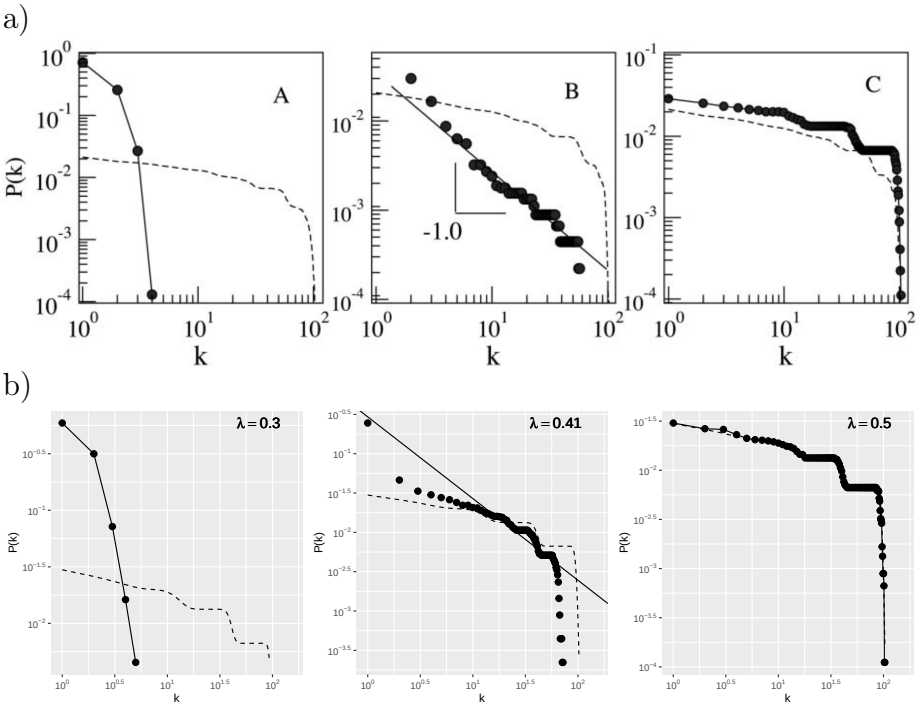


Figure 4.4: Normalized signal frequency $P(k)$ versus rank k. The dashed lines show the distribution obtained when from a G_{n,m,e^*} graph where e^* is the number of connections in these optimal configurations. In both cases the distribution of B is still consistent with human language, $\alpha = 1$. Averages over 30 replicas, $n = m = 150$, the initial graph is $G_{n,m,10/n}$, the number of mutations on each iteration follows a binomial distribution with average 1.75, the optimization stops after $2nm$ mutations that do not improve the cost function.
Above: Figure 3 from [13]
Below: Recreation of that same figure using the open source tool created for this thesis.

4.2 Results for current models

Here new results obtained from the model are presented. Firstly, and as a way to for an intuition of the kinds of graphs that these models generate, Section 4.2.1 presents the optimal graphs for different initial conditions for both models. Section 4.2.2 shows the rest of the obtained results. This section focuses first on plots of the values obtained for the information theoretical measures of the optimal graphs for various value of λ in the cost function. As a reminder, λ controls the weight of either the entropy or the mutual information in the cost function. Statistical measures for select values of λ are then shown. These measures show how various linguistic laws appear in the results. Several kinds of initial graphs are shown throughout this section.

- Random: The initial graph is a $G_{n,m,3/n/m}$ random graph.
- Single link: The initial graph contains a single link between a word and a meaning.
- One-to-one: The initial graph is a bijection connecting words and meanings one to one
- Complete: The initial graph is a complete bipartite graph where every word connects to every meaning and every meaning to every word.

All results in this section perform two mutations on each iteration of the optimization algorithm, which stops with the *weak* stop condition. See Section 3.2 for more information on the optimization process.

4.2.1 Graph visualization

Here various graphs are plotted. These plots all correspond to graphs of size $n = m = 60$ and $\phi = 1$. Unlinked meanings are also allowed in all graphs. It is interesting to see the different shapes the graph takes for different values of λ.

For the first model, four graphs are shown.

- Random initial graph, Figure 4.5.
- Single link initial graph, Figure 4.6.
- One-to-one initial graph, Figure 4.7.
- Complete initial graph, Figure 4.8.

For the second model, the *a priori* probability π follows a uniform distribution. Four additional graphs are shown.

- Random initial graph, Figure 4.9
- Single link initial graph, Figure 4.10
- One-to-one initial graph, Figure 4.11
- Complete initial graph, Figure 4.12

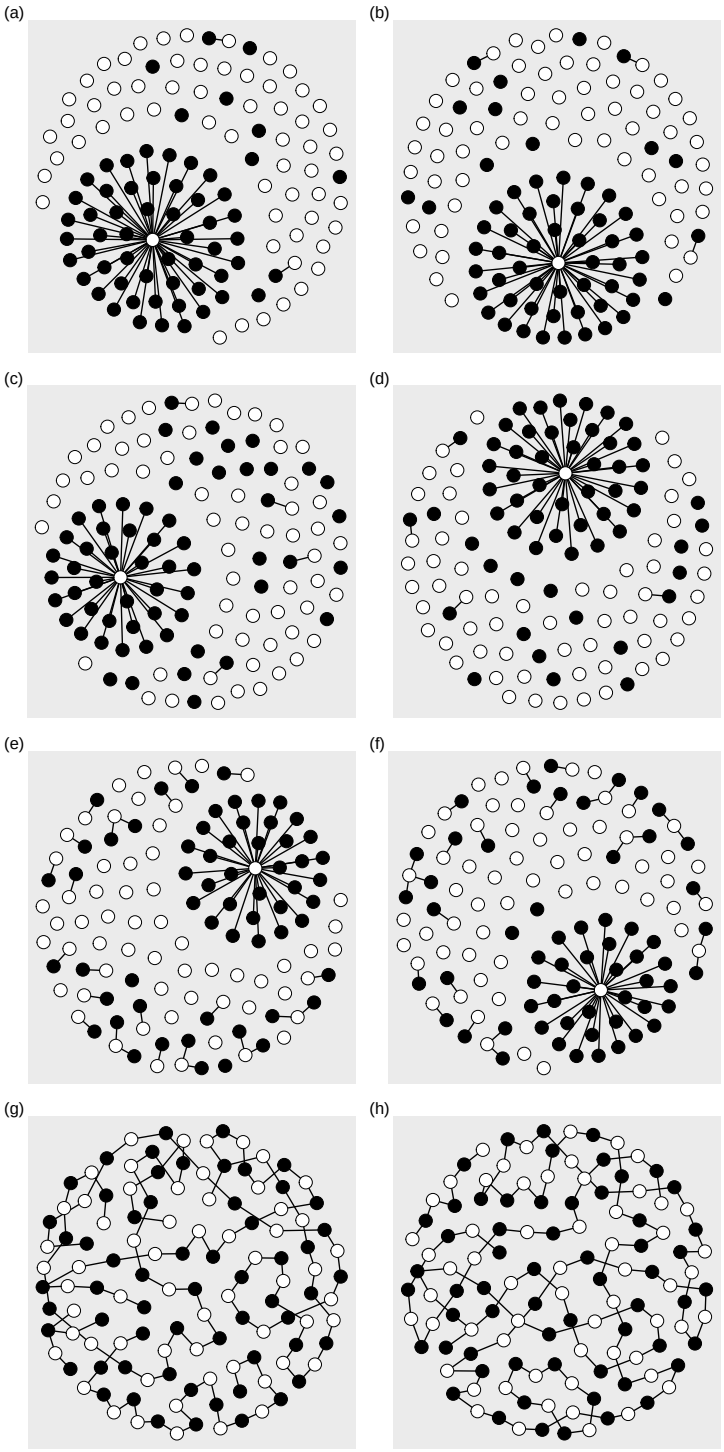


Figure 4.5: a

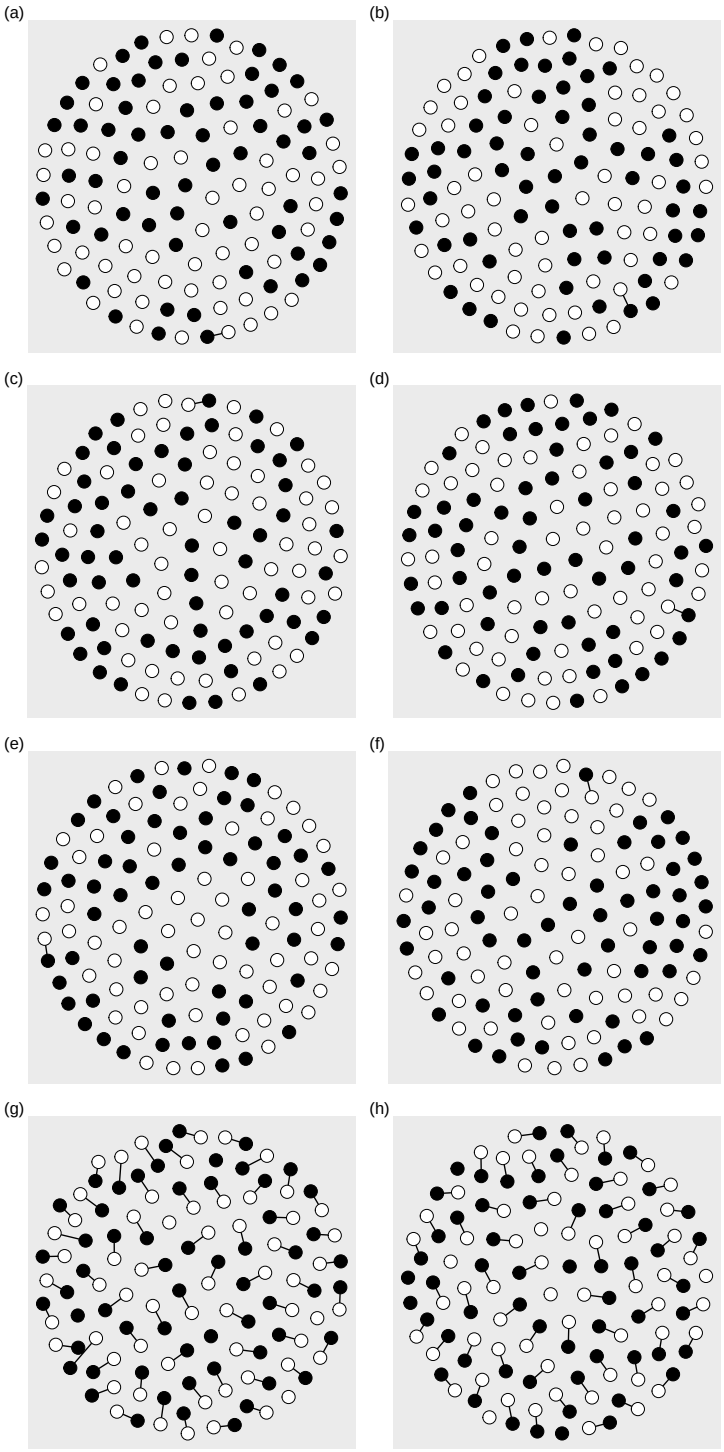


Figure 4.6: a

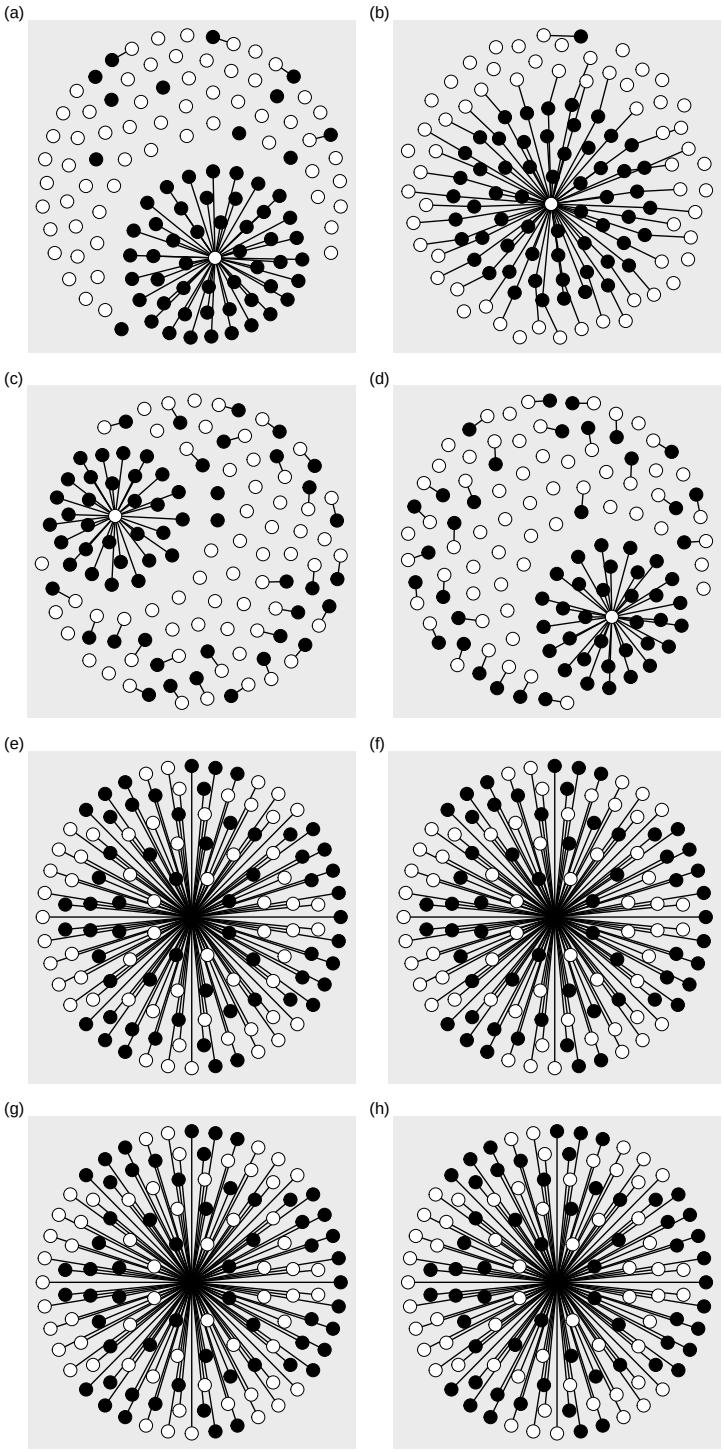


Figure 4.7: a

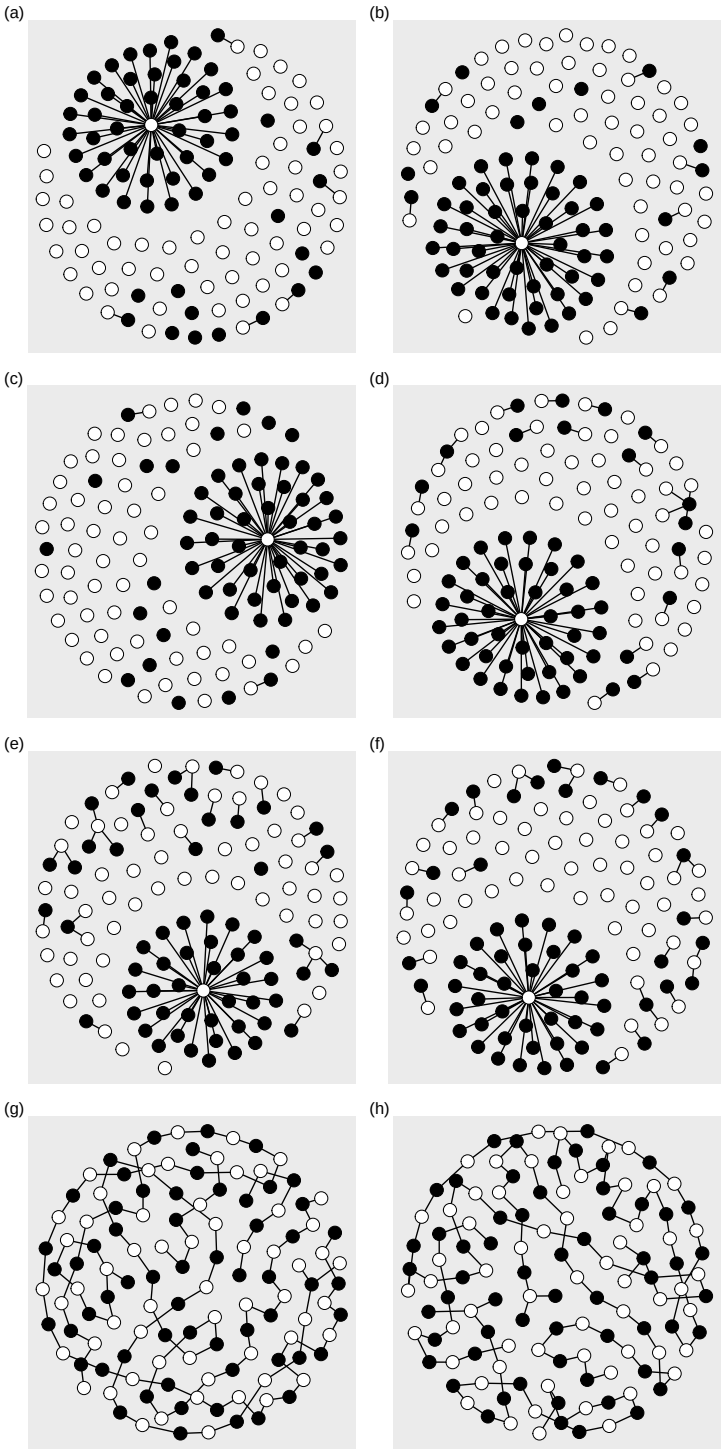


Figure 4.8: a

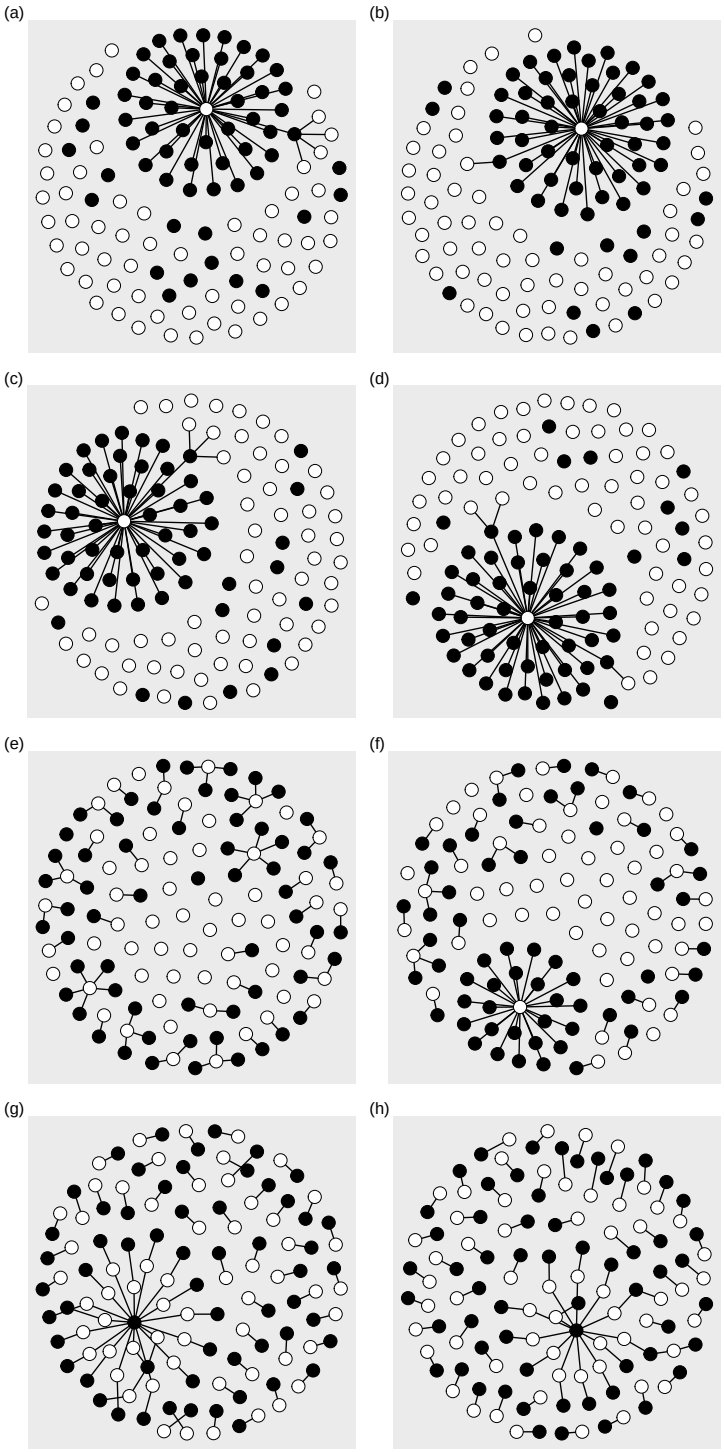


Figure 4.9: a

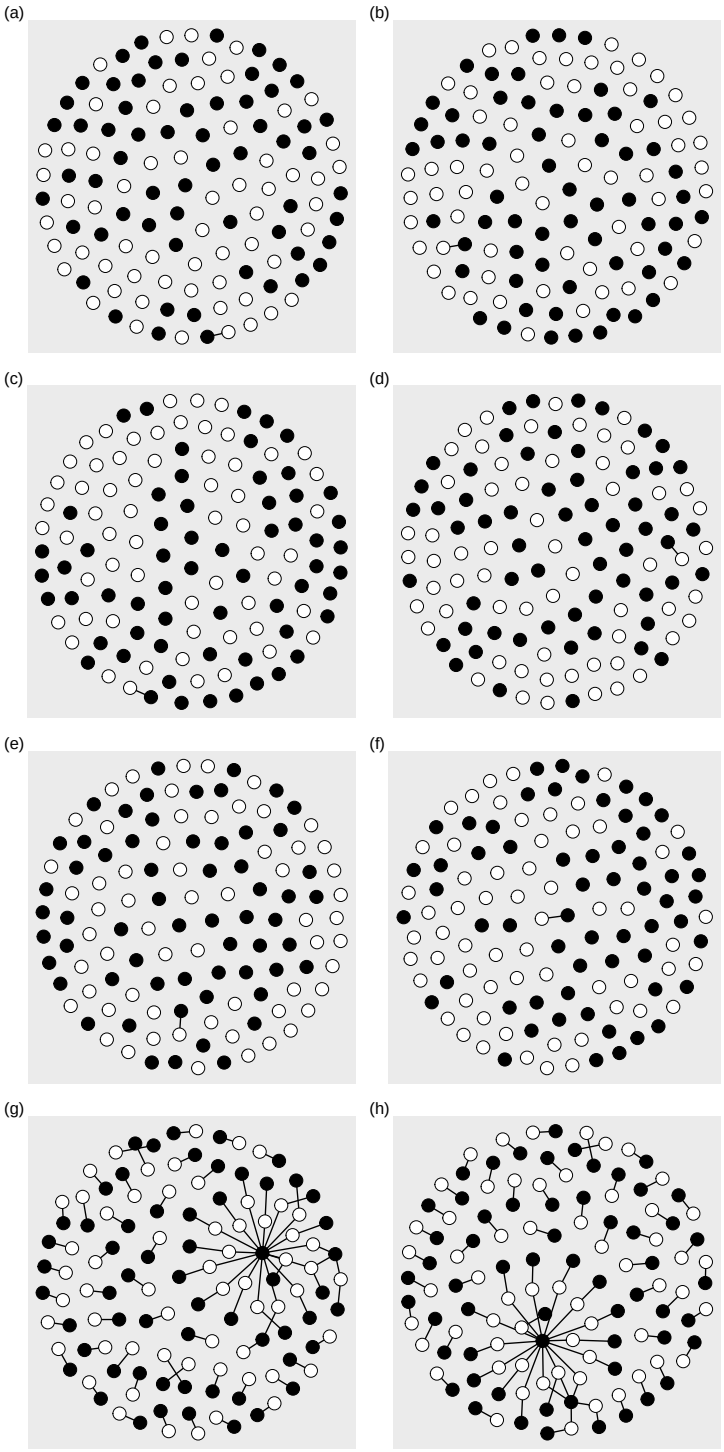


Figure 4.10: a

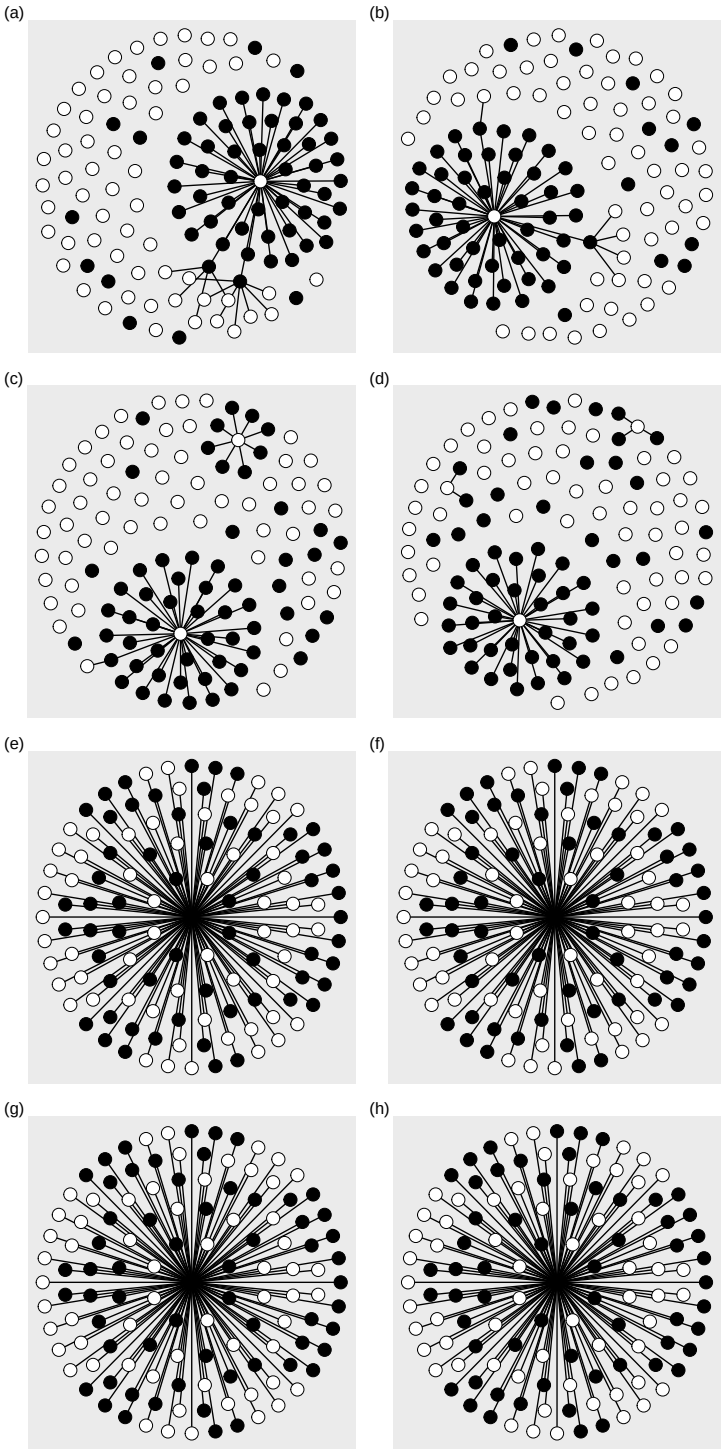


Figure 4.11: a

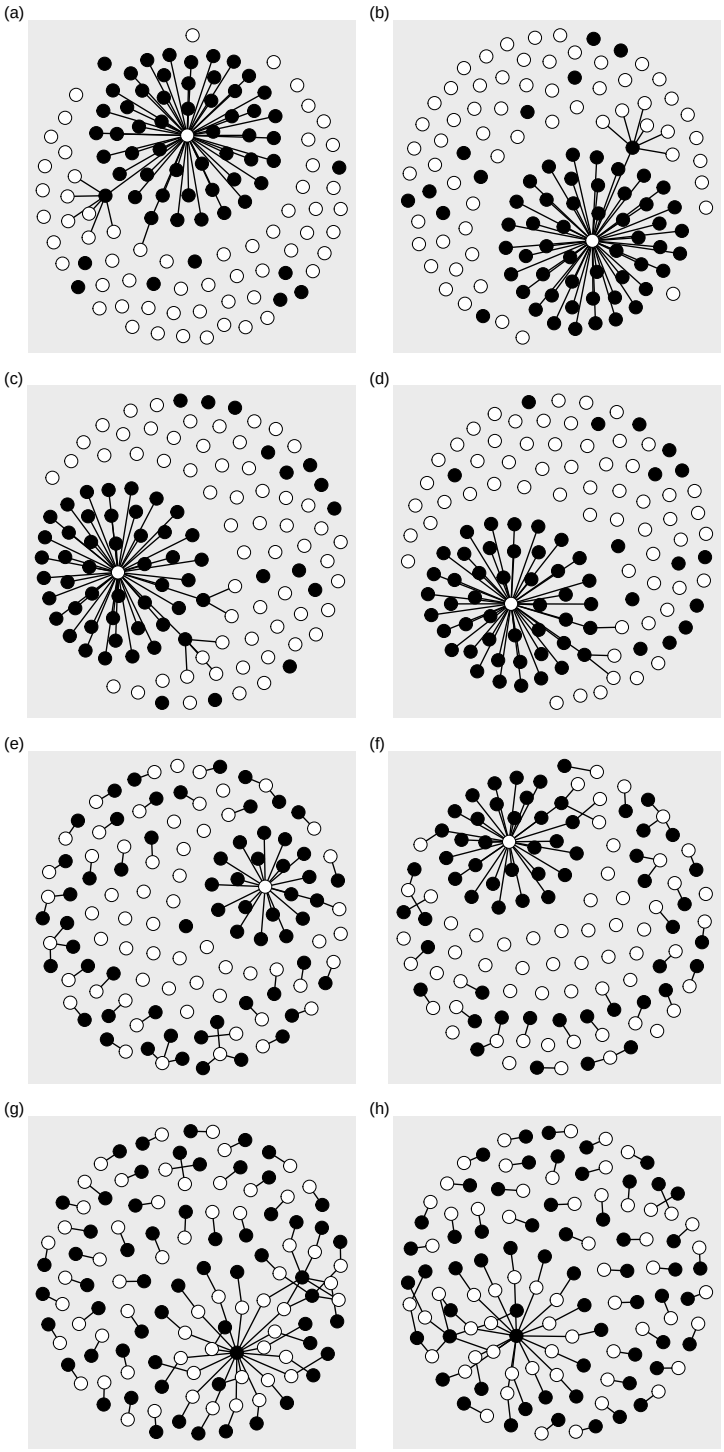


Figure 4.12: a

plot	α	k
a	1.5131590	0.4415009
b	0.6406267	0.0016491
c	1.4972982	170.7165580
d	-0.9992804	403.8279052
f	1.4972982	170.7165580
g	0.6725912	0.0750000

Table 4.1: Table showing the exponent (α) and the factor (k) of the power laws fitted in Figure 4.18 for each of the subfigures. The power law follows the formula $y = kx^{-\alpha}$.

4.2.2 Information theoretic and statistical measures

This section shows the data obtained from executing simulations with various sets of parameters. All simulations are run with a graph size of $n = m = 400$. Every experiment in this section is the result of averaging at least 20 realizations. Most experiments consist of 100 realizations. Some had to be reduced to 20 realizations due to the total running time and are marked as such. Results for $\phi = 0$ and $\phi = 1$ are shown. Three different initial conditions are tested: random bipartite graph, single link and one-to-one. For the external model only results for π following a uniform probability distribution are shown. The chosen λ value for which curves are fitted is the one that's qualitatively closer to a power law.

For the internal model with $\phi = 0$, Figures 4.13, 4.14 and 4.15 show the information theoretic measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the random graph, single link and one-to-one initial graphs respectively. It can be seen how there is a phase transition when $\lambda \approx 0.5$. There is a change in the behavior before the point where the phase changes in both the random bipartite and the one-to-one initial conditions but not in the single link.

Figures 4.16 and 4.17 (random and one-to-one initial conditions respectively) show statistical measures of select values of λ, with Figures 4.18 and 4.19 (random and one-to-one respectively) showing the fitting of the curve to a power law for a single select value of λ. It can be seen that a power law (linear in log-log scale) appears in the plots. Tables 4.1 and 4.2 (random and one-to-one respectively) showing the values of the regression exponent and factor. The single link initial condition is not plotted for select values of λ as it fails to evolve beyond the single link state.

For the internal model but $\phi = 1$, Figures 4.20, 4.21 and 4.22 show the information theoretic measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the random graph, single link and one-to-one initial graphs respectively. As with the $\phi = 0$ variant, there is a phase transition when $\lambda \approx 0.5$. The changes before that point, however, are much less pronounced.

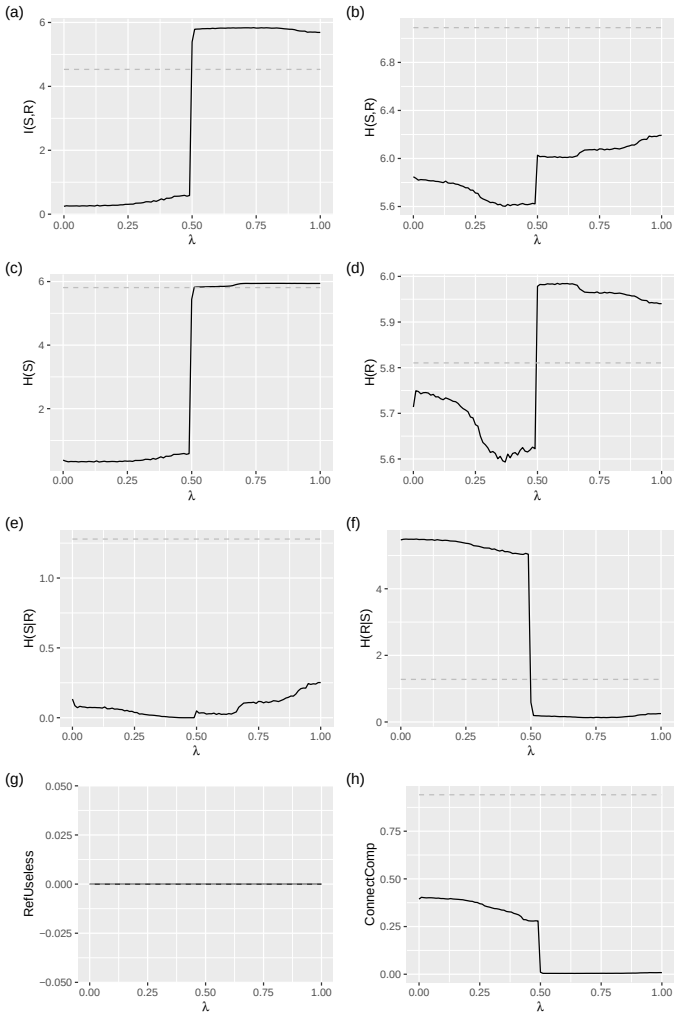


Figure 4.13: Information theoretic measures for the optimized graph. The graph follows the equations of the internal model with $\phi = 0$ and the initial condition of the optimization process is a random bipartite graph $G_{n,m,3/n/m}$. Disconnected meanings are allowed. On the x axis of each subfigure is the parameter λ of the optimization process, ranging from 0 to 1. On the y of the subfigures is: the mutual information between words and meanings (a), the joint entropy between words and meanings (b), the entropy of words (c), the entropy of meanings (d), the conditional entropy of words given the meanings (e), the conditional entropy of the meanings given the words (f), the number of referentially useless words (g) and the proportion of the largest connected component of the graph (h). A word s_i is referentially useless if it is connected to at least one meaning and for each meaning r_j it is connected to, $p(s_i, r_j) \leq p(s_i) \cdot p(r_j)$. Averages over 100 realizations.

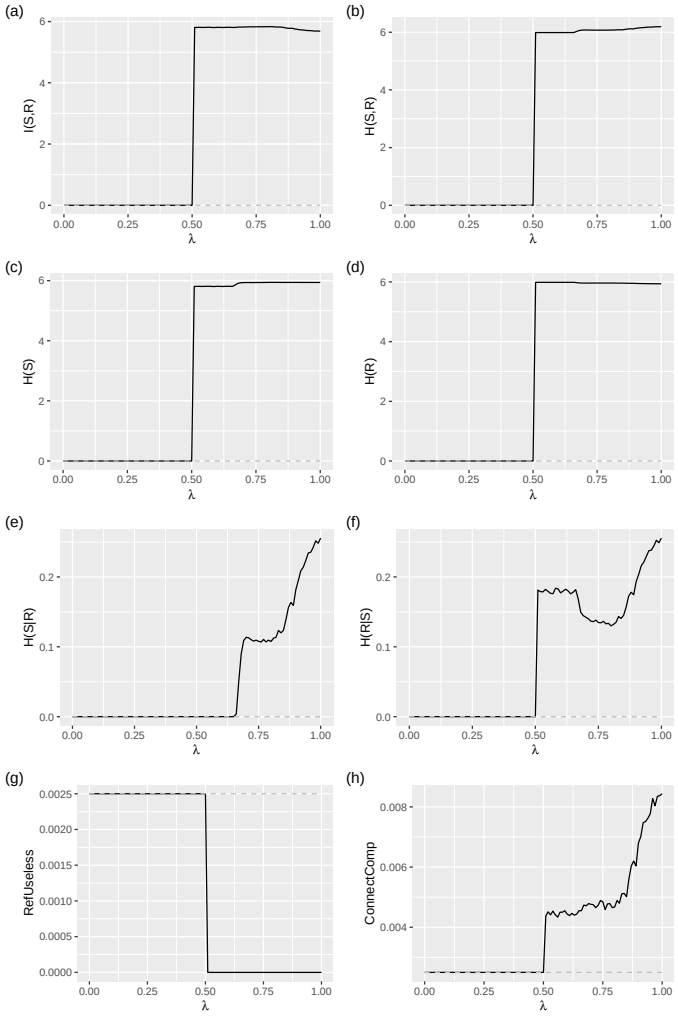


Figure 4.14: Same information as in Figure 4.13 but for a single link as the initial condition.

plot	α	k
a	4.0992847	2.3970141
b	0.2466051	0.0030639
c	3.9365300	577.6844467
d	-0.9673668	271.7315918
f	3.9365300	577.6844467
g	0.2731698	0.0125000

Table 4.2: Table showing the exponent and factor of the power laws fitted in Figure 4.19.

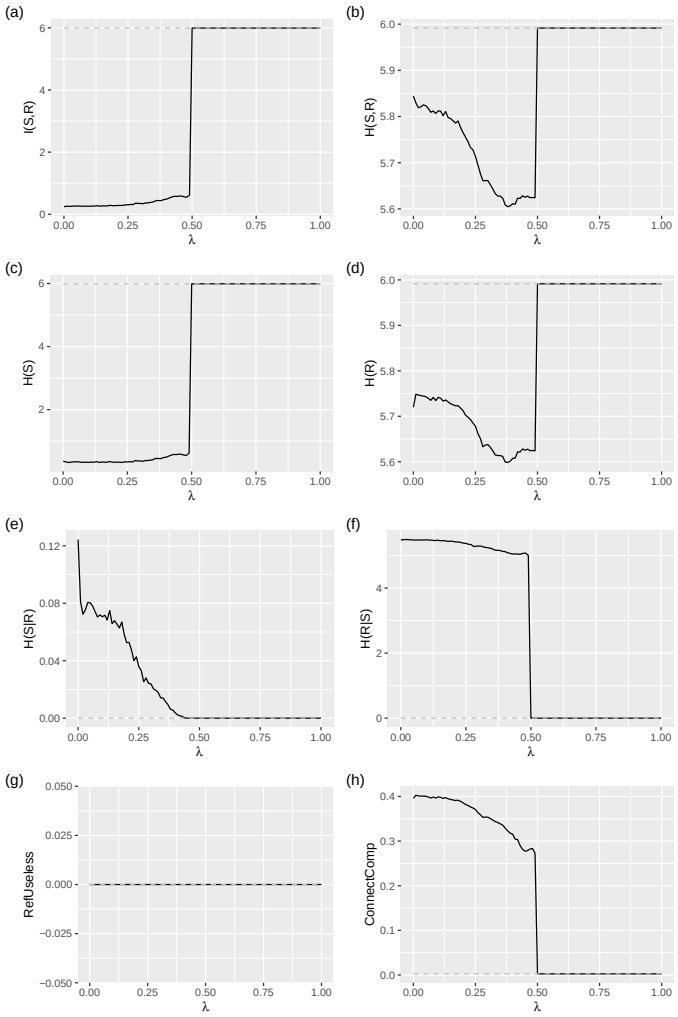


Figure 4.15: Same information as in Figure 4.13 but for one to one connections between signals and meanings as the initial condition.

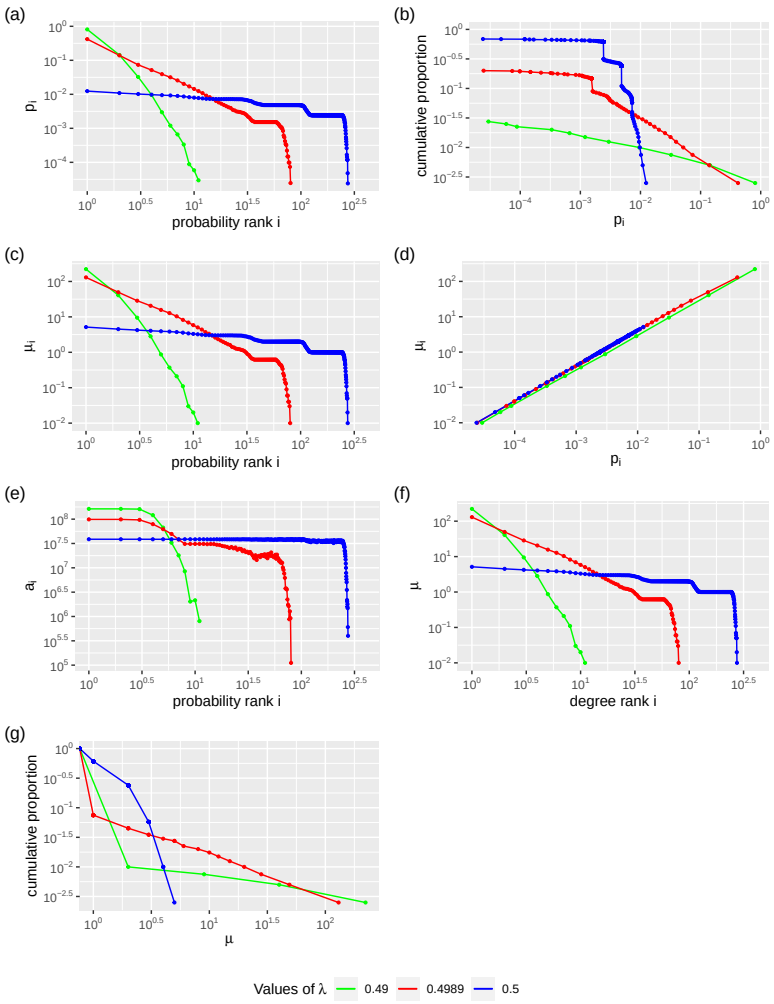


Figure 4.16: Statistical measures of the optimized graph for select values of λ. The graph follows the equations of the internal model with $\phi = 0$ and the initial condition of the optimization process is a random bipartite graph $G_{n,m,3/n/m}$. Disconnected meanings are allowed. The green line corresponds to $\lambda = 0.49$, the blue line to $\lambda = 0.5$ and the red line to $\lambda = \lambda^*$. The value of λ^* is chosen qualitatively and in this case $\lambda^* = 0.4989$. Figure (a) shows the probability (or frequency) of a word as a function of its probability rank. Figure (b) shows the cumulative proportion of the frequency of words. Figure (c) shows the number of meanings of a word as a function of its probability rank. Figure (d) shows the number of meanings of a word as a function of its probability. Figure (e) shows age of a word as a function of its probability rank. Figure (f) shows the number of meanings of a word as a function of its degree rank. Figure (g) shows the cumulative proportion of the number of meanings of words. All figures are in a log-log scale. Averages over 100 realizations.

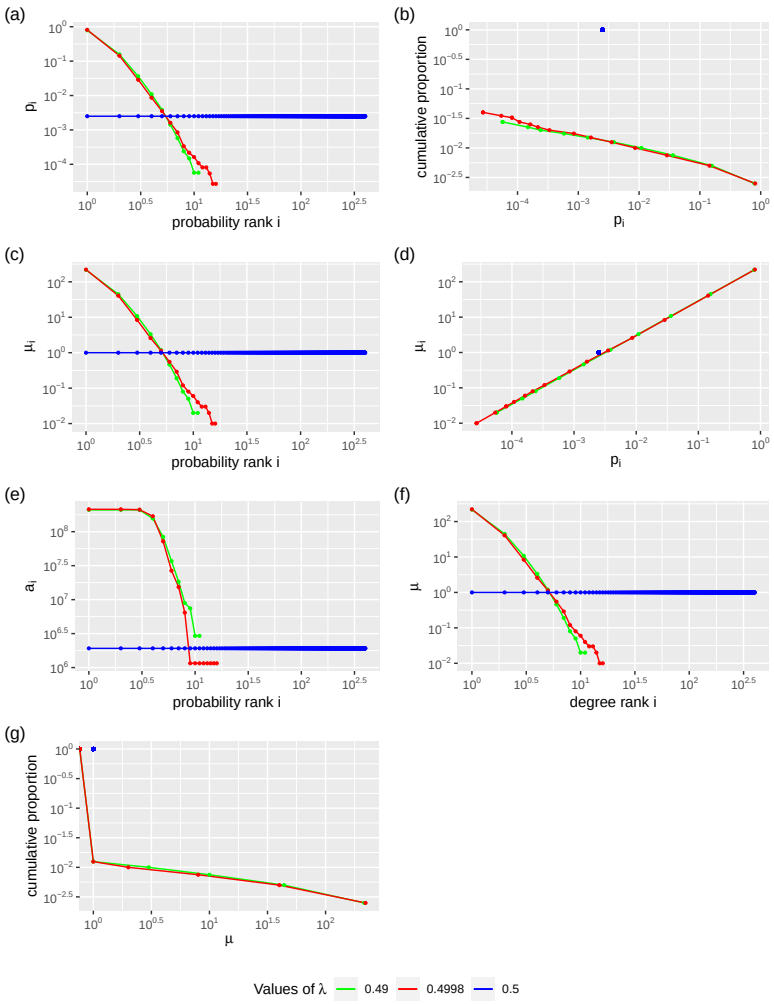


Figure 4.17: Same information as in Figure 4.16 but the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4989$

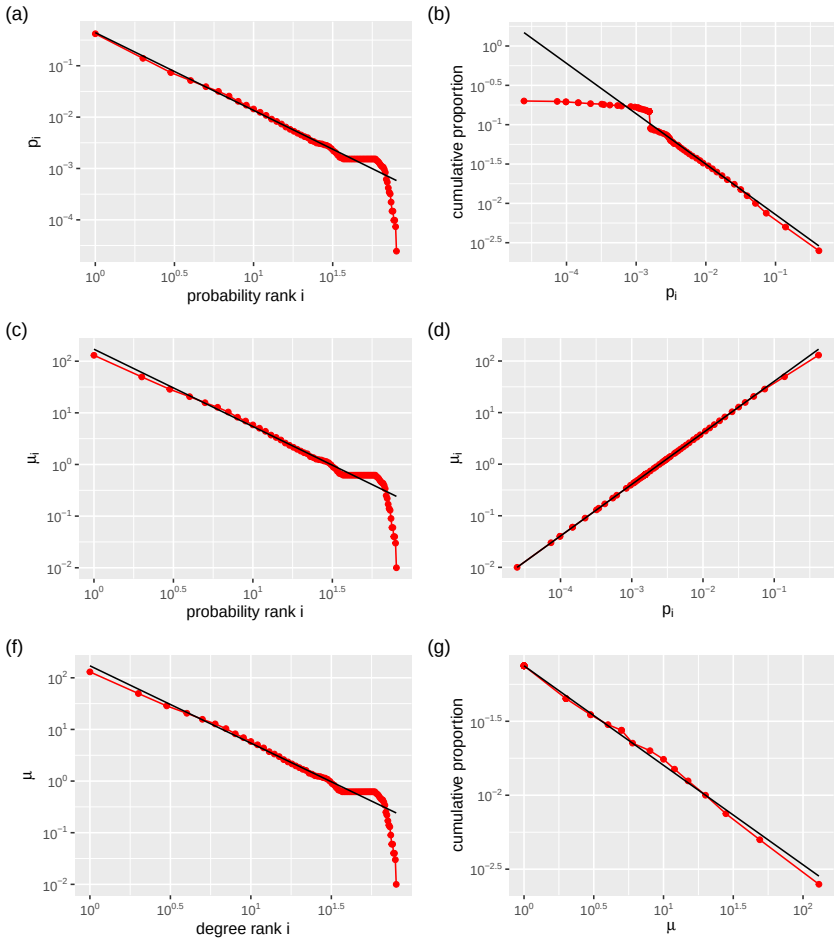


Figure 4.18: Statistical measures of the optimized graph for a select value of $\lambda = \lambda^*$. The graph follows the equations of the internal model with $\phi = 0$ and the initial condition of the optimization process is a random bipartite graph $G_{n,m,3/n/m}$. Disconnected meanings are allowed. $\lambda^* = 0.4989$. The black line indicates the values predicted by the Theil-Sen liniear regression. The red line indicates the actual values in the graph. Figure (a) shows the probability (or frequency) of a word as a function of its probability rank. Figure (b) shows the cumulative proportion of the frequency of words. Figure (c) shows the number of meanings of a word as a function of its probability rank. Figure (d) shows the number of meanings of a word as a function of its probability. Figure (f) shows the number of meanings of a word as a function of its degree rank. Figure (g) shows the cumulative proportion of the number of meanings of words. Figure (e), which should show the age of a word as a function of its probability rank, is omitted as it never followed a power law. All figures are in a log-log scale. Table 4.1 shows the values of the exponent and the factor of the fitted power law.

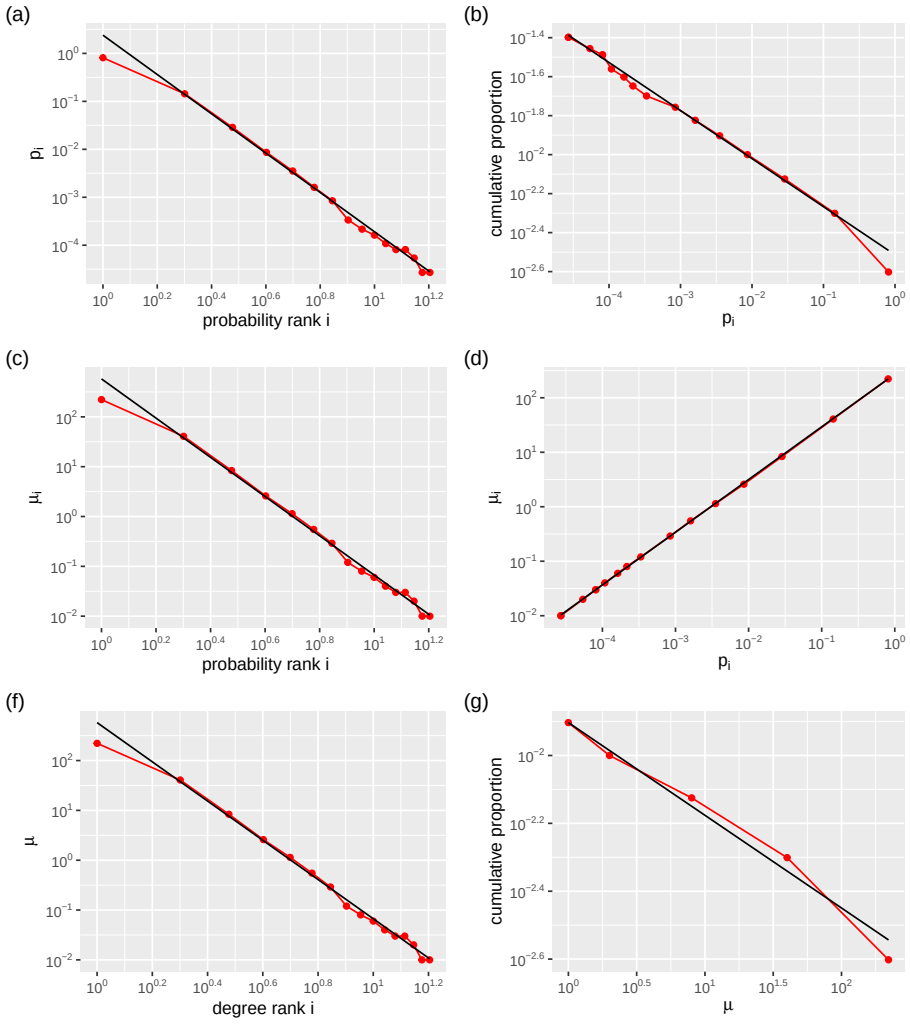


Figure 4.19: Same information as in Figure 4.18 but the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4989$. Table 4.2 shows the values of the exponent and the factor of the fitted power law.

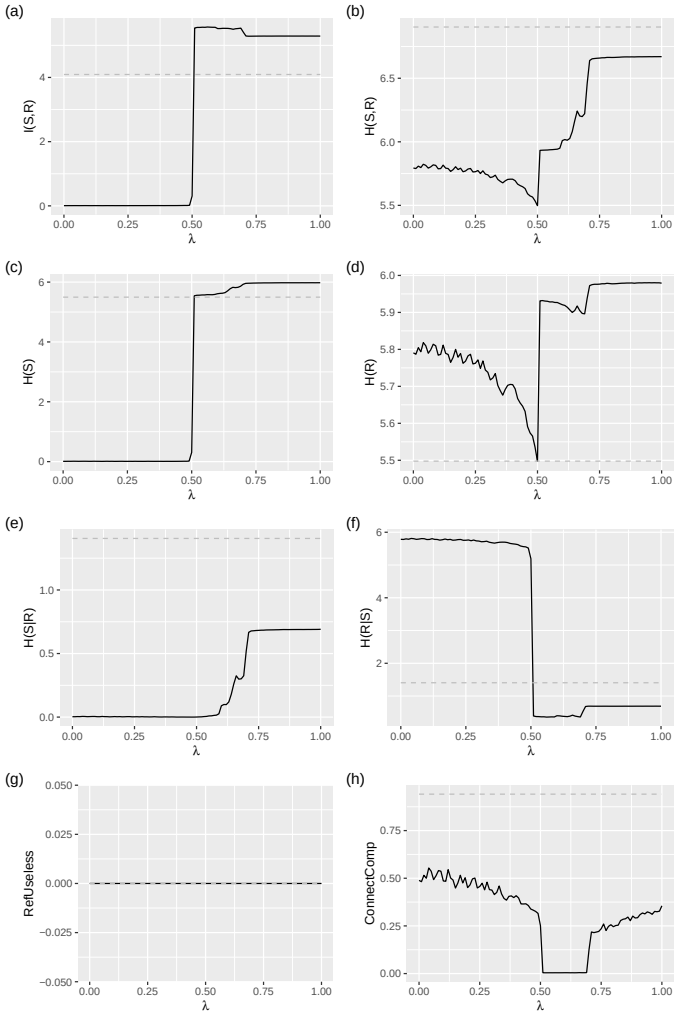


Figure 4.20: Same information as in Figure 4.13 but with $\phi = 1$.

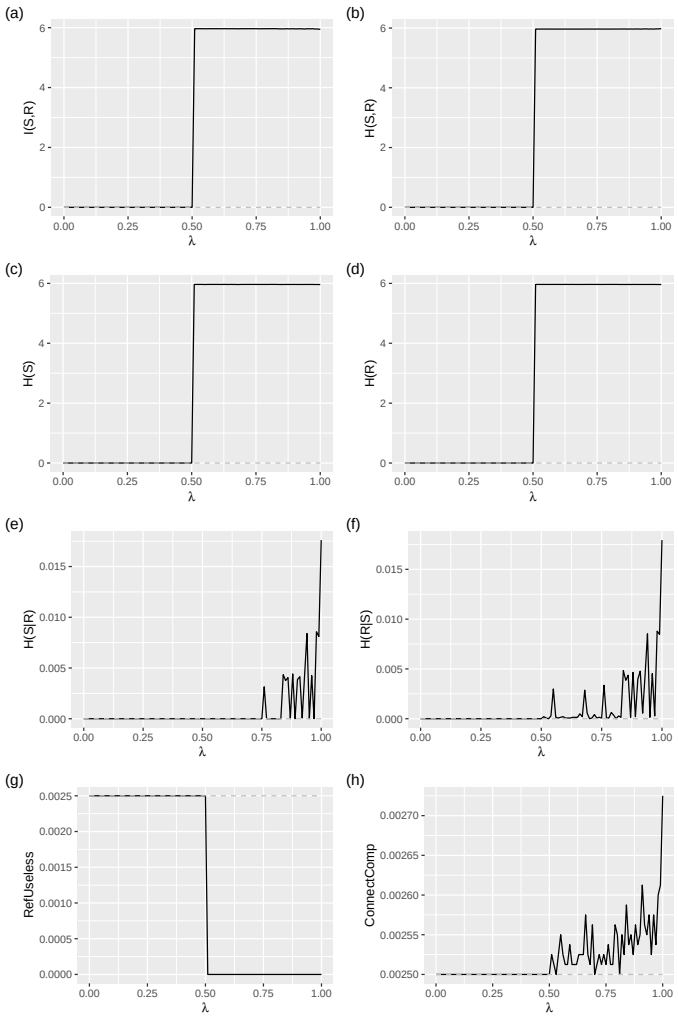


Figure 4.21: Same information as in Figure 4.13 but with $\phi = 1$ and for a single link as the initial condition.

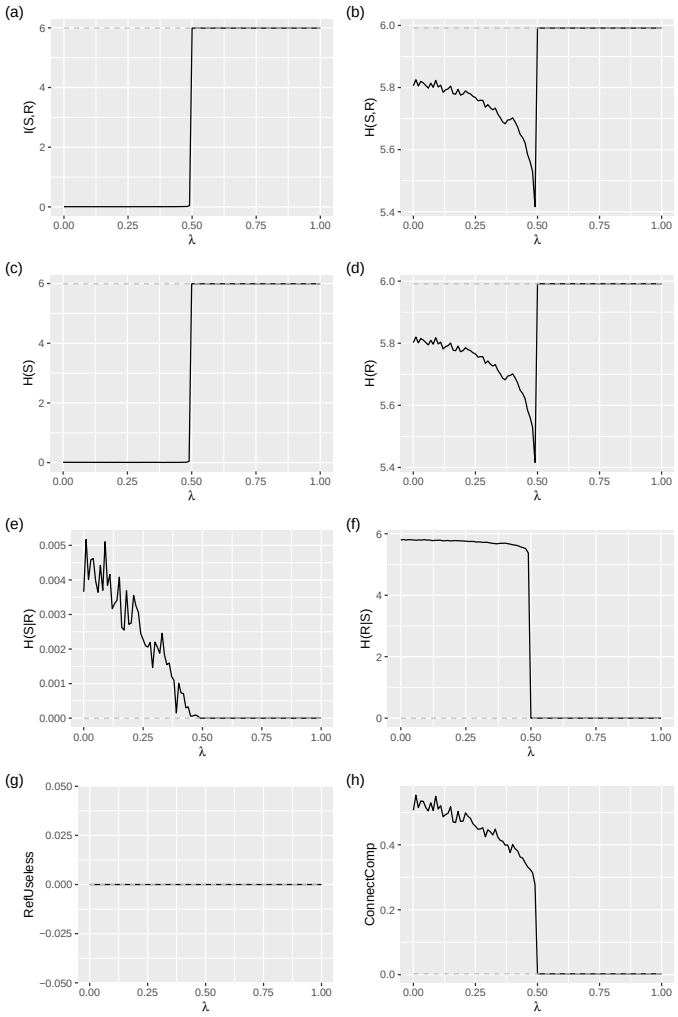


Figure 4.22: Same information as in Figure 4.13 but with $\phi = 1$ and for one to one connections between signals and meanings as the initial condition.

Figures 4.23 and 4.24 (random and one-to-one initial conditions respectively) show statistical measures of select values of λ, with Figures 4.25 and 4.26 (random and one-to-one respectively) showing the fitting of the curve to a power law for a single select value of λ. However, in this case it's hard to say that the curves follow a power law. Nevertheless, Tables 4.3 and 4.4 (random and one-to-one respectively) show the values of the regression exponent and factor. The single link initial condition is not plotted for select values of λ as it fails to evolve beyond the single link state.

plot	α	k
a	0.2871434	0.0000468
b	2.5493723	0.0000000
c	0.3072861	2.8668125
d	-1.0876280	149669.1453764
f	0.3072861	2.8668125
g	0.5944739	0.0650000

Table 4.3: Table showing the exponent and factor of the power laws fitted in Figure 4.25

plot	α	k
a	0.2433438	0.0000524
b	2.2902299	0.0000000
c	0.3100715	3.2784604
d	-1.2166546	501643.7856344
f	0.3100715	3.2784604
g	0.4063538	0.0225000

Table 4.4: Table showing the exponent and factor of the power laws fitted in Figure 4.26

For the external model with $\phi = 0$ Figures 4.27, 4.28 and 4.29 show the information theoretic measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the random graph, single link and one-to-one initial graphs respectively. Appendix C.1 shows the same figures but with disconnected meanings disallowed.

Figures 4.30 and 4.31 (random and one-to-one respectively) show statistical measures of select values of λ, with Figures 4.32 and 4.33 (random and one-to-one respectively) showing the fitting of the curve to a power law for a single select value of λ. A power law is appreciated in some of the plots but not all of them. Tables 4.5 and 4.6 (random and one-to-one respectively) show the values of the regression exponent and factor. As with others, the single link initial condition is not plotted for select values of λ as it fails to evolve beyond the single link state. Appendix C.1 shows the same figures but with disconnected meanings disallowed.

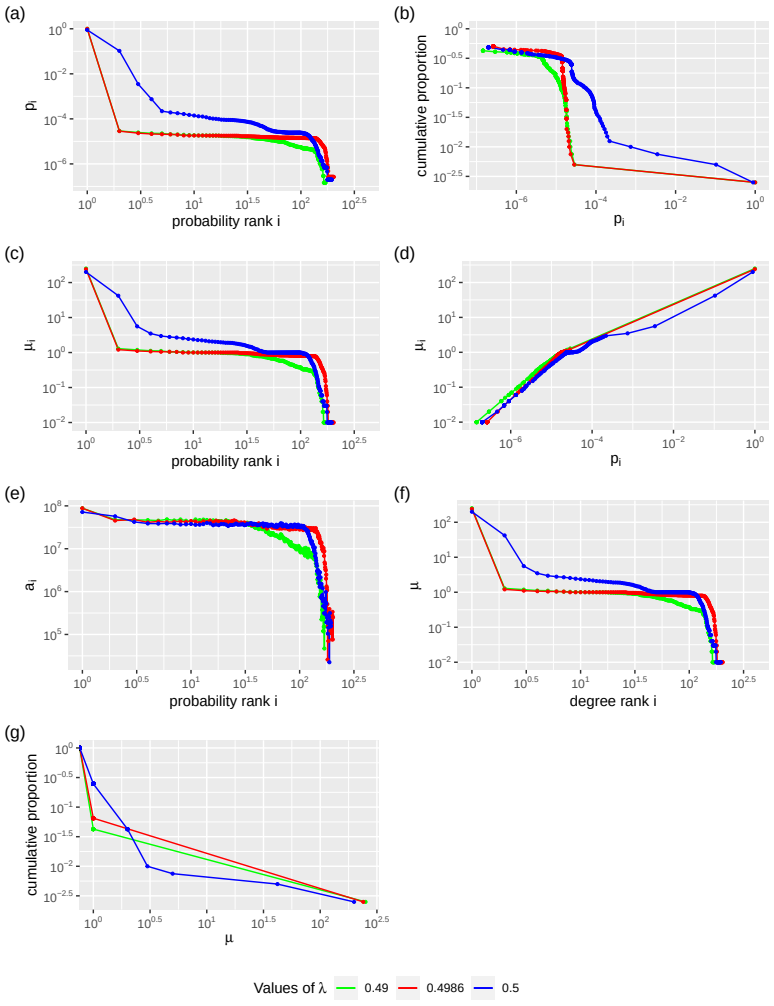


Figure 4.23: Same information as in Figure 4.16 but $\phi = 1$. $\lambda^* = 0.4993$

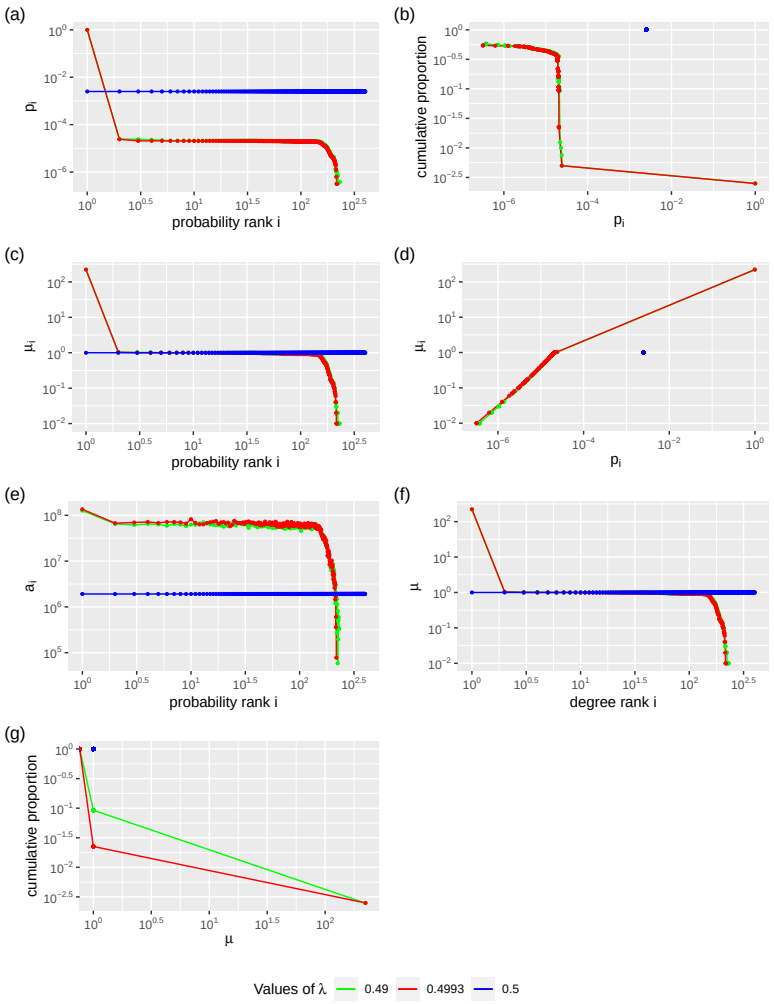


Figure 4.24: Same information as in Figure 4.16 but $\phi = 1$ and the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4986$

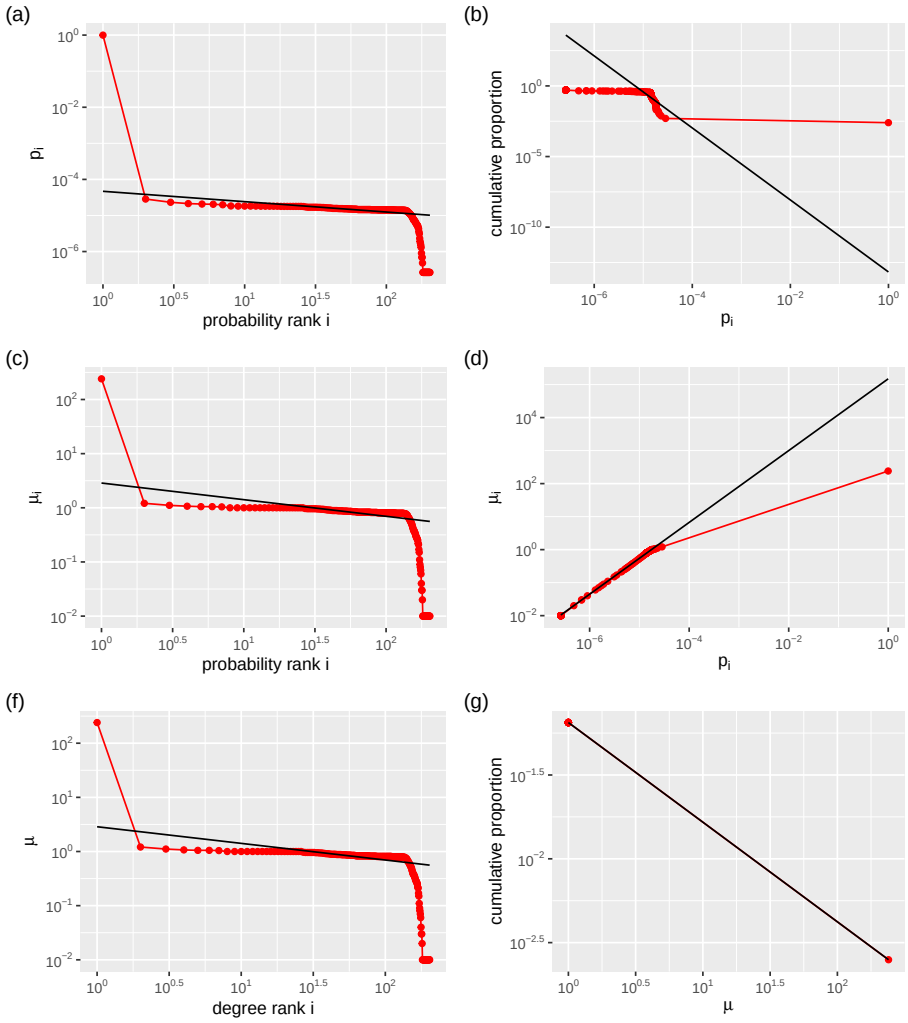


Figure 4.25: Same information as in Figure 4.18 but $\phi = 1$. $\lambda^* = 0.4986$. Table 4.3 shows the values of the exponent and the factor of the fitted power law.

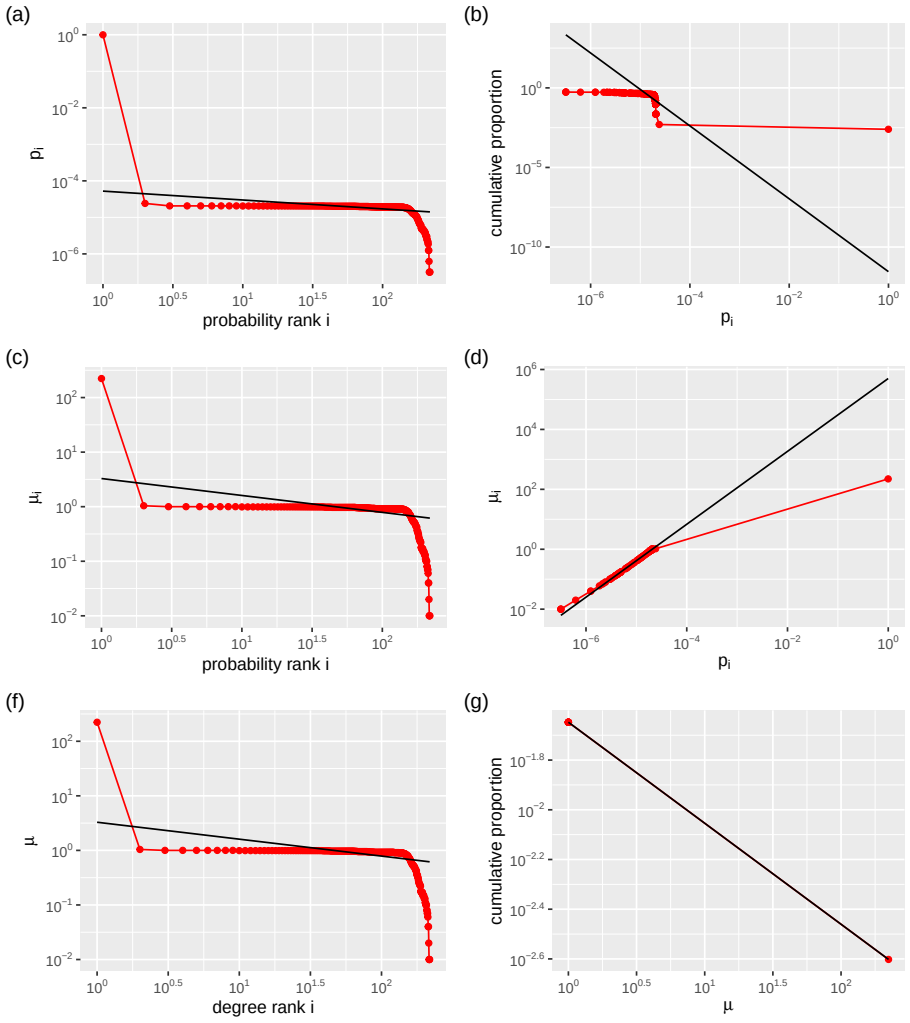


Figure 4.26: Same information as in Figure 4.18 but $\phi = 1$ and the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4993$. Table 4.4 shows the values of the exponent and the factor of the fitted power law.

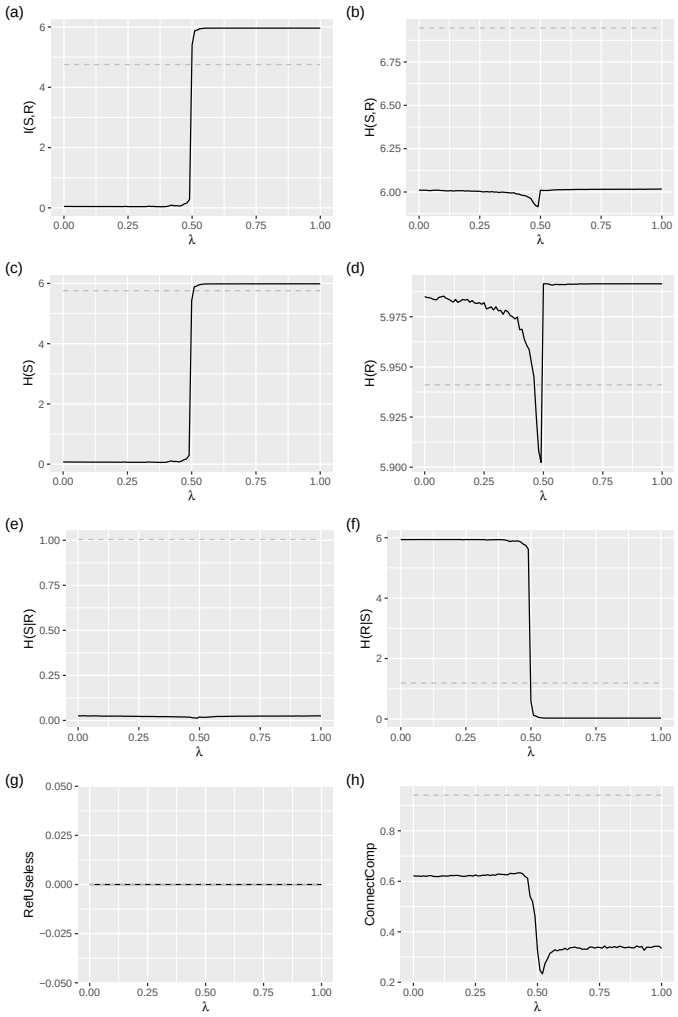


Figure 4.27: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution.

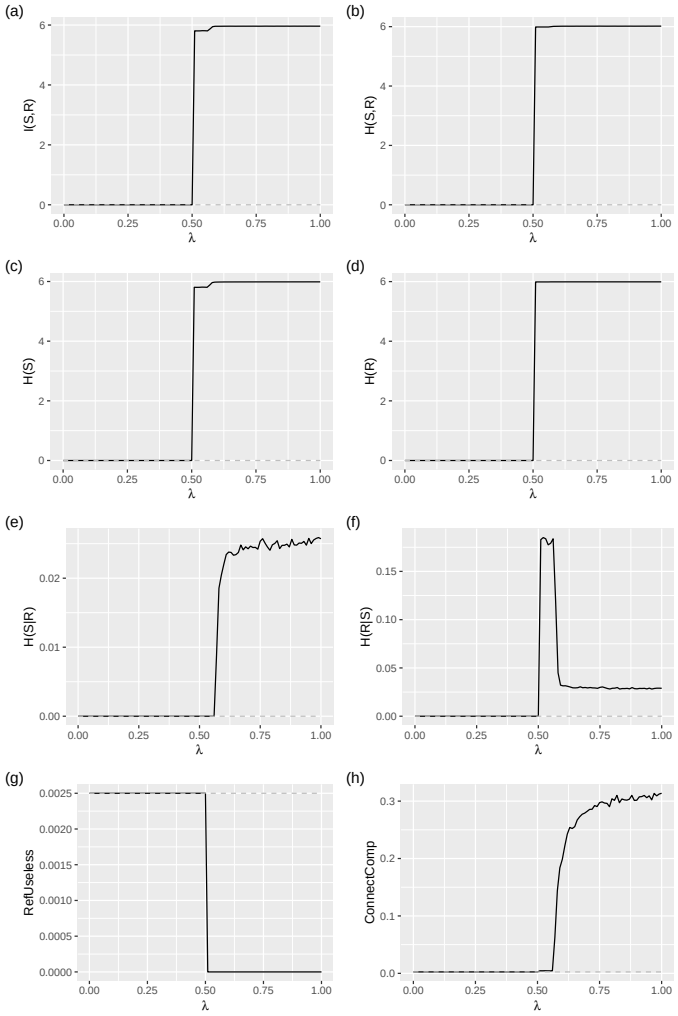


Figure 4.28: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution and the initial condition is a single link

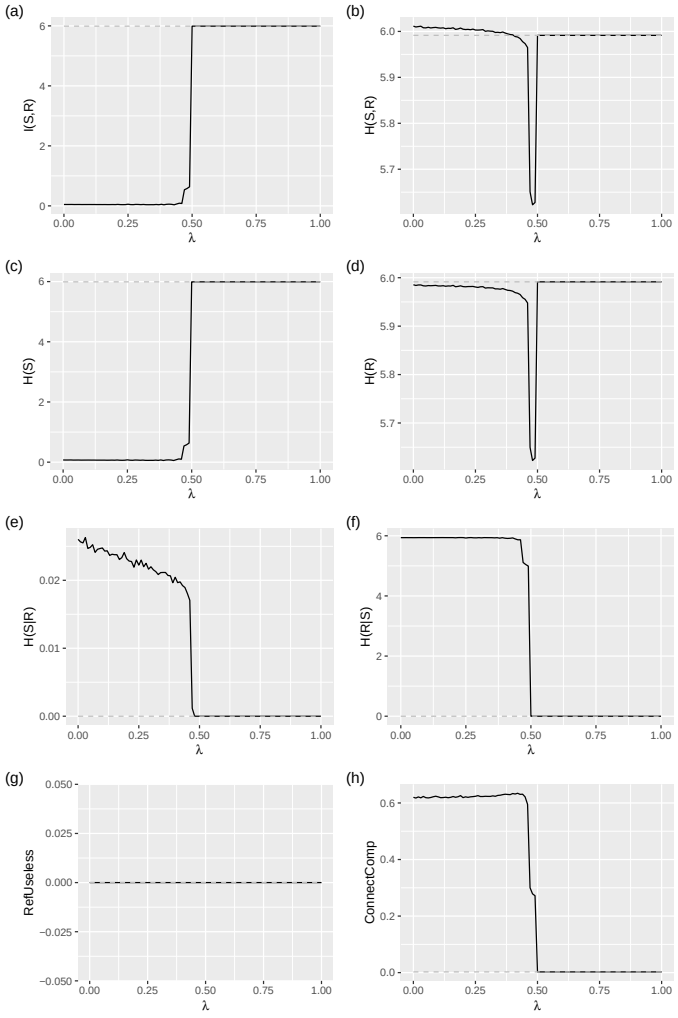


Figure 4.29: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution and the initial condition is one to one connections between signals and meanings.

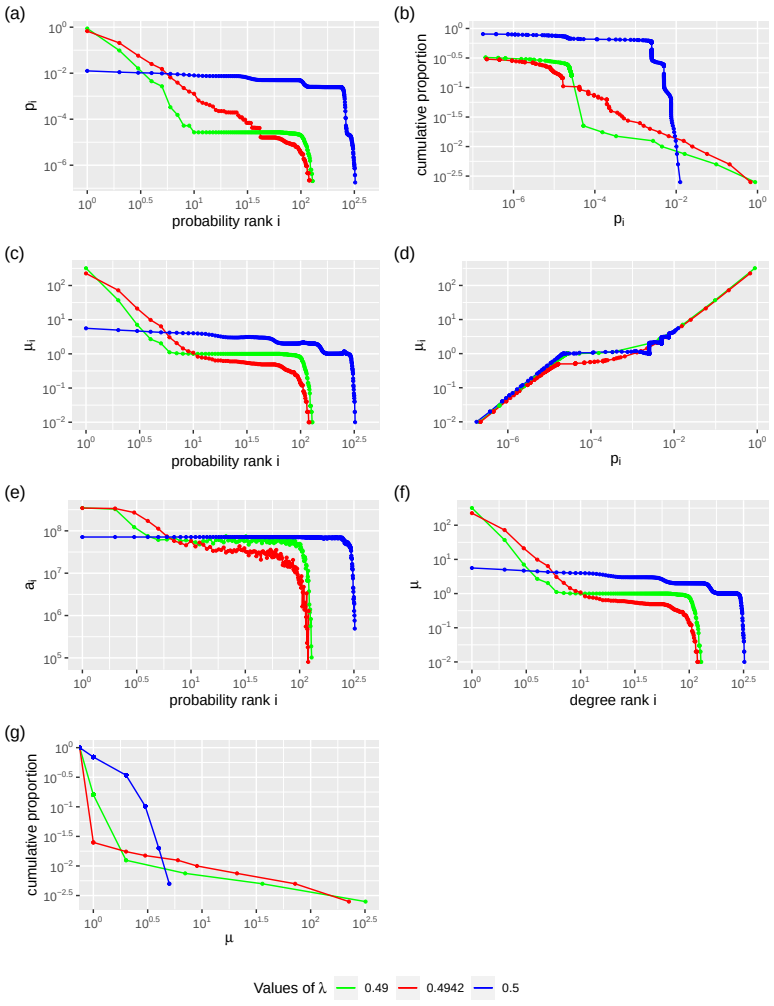


Figure 4.30: Same information as in Figure 4.16 but the graph follows the equations of the external model with π following a uniform distribution. $\lambda^* = 0.4942$

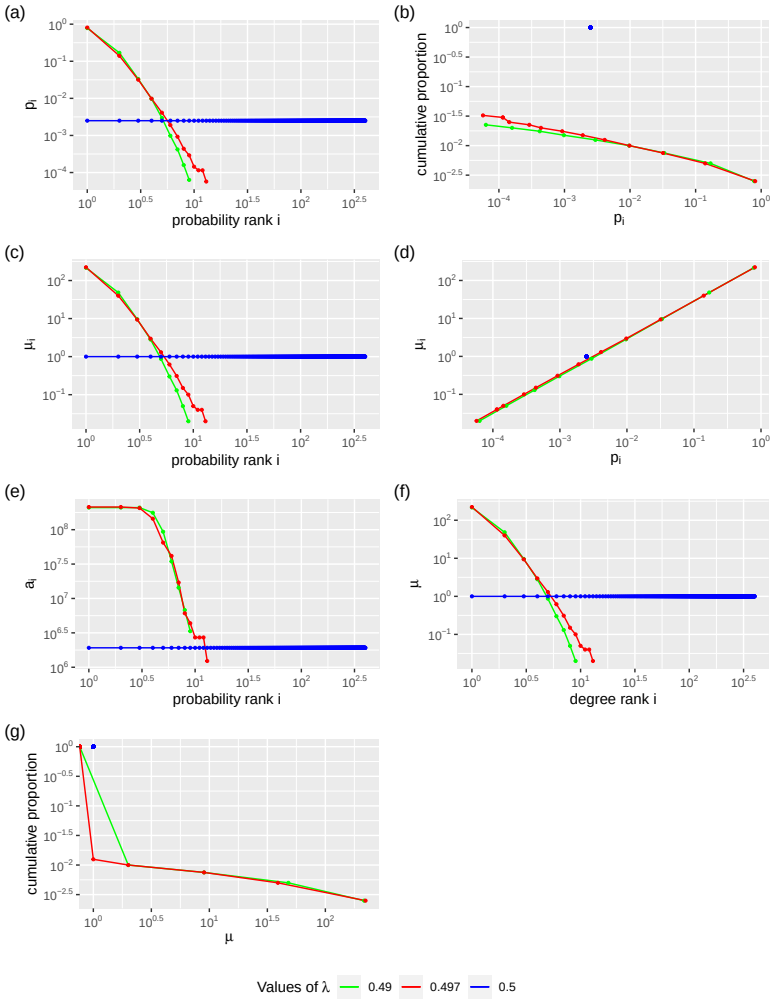


Figure 4.31: Same information as in Figure 4.16 but the graph follows the equations of the external model with π following a uniform distribution and the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4970$

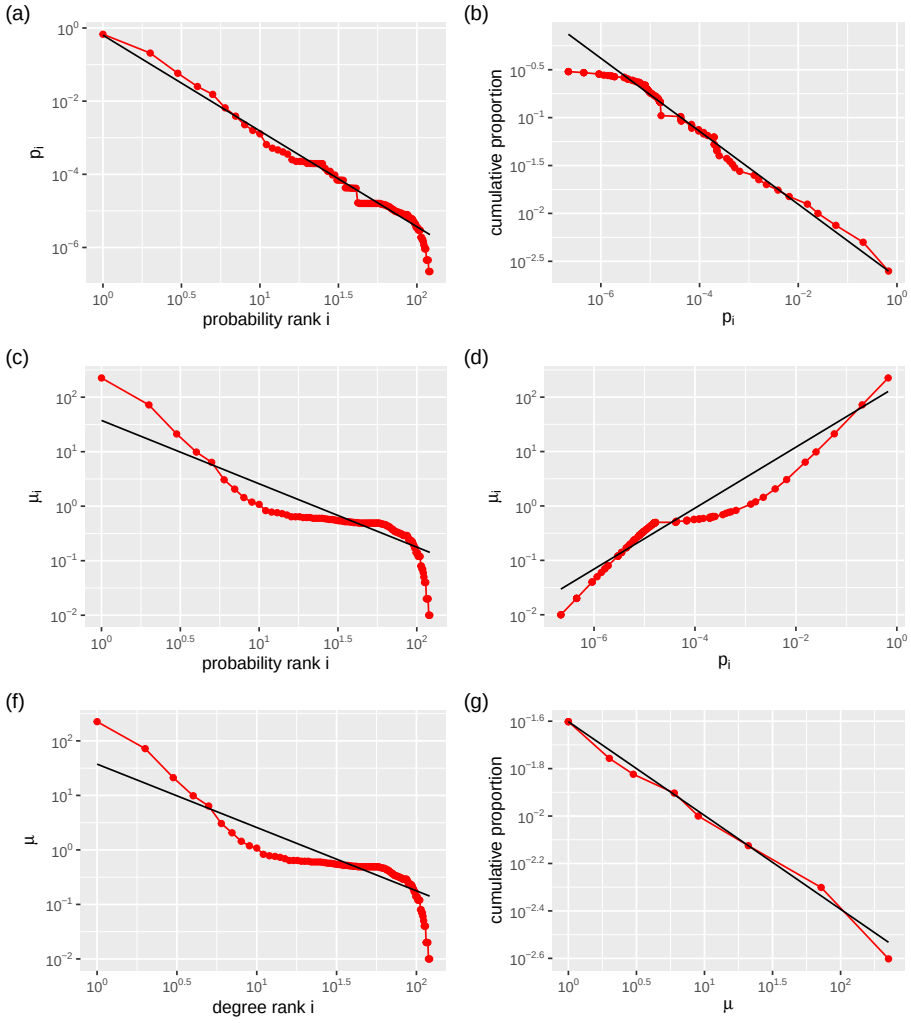


Figure 4.32: Same information as in Figure 4.18 but the model follows the equations of the external model with π following a uniform distribution. $\lambda^* = 0.4942$. Table 4.5 shows the values of the exponent and the factor of the fitted power law.

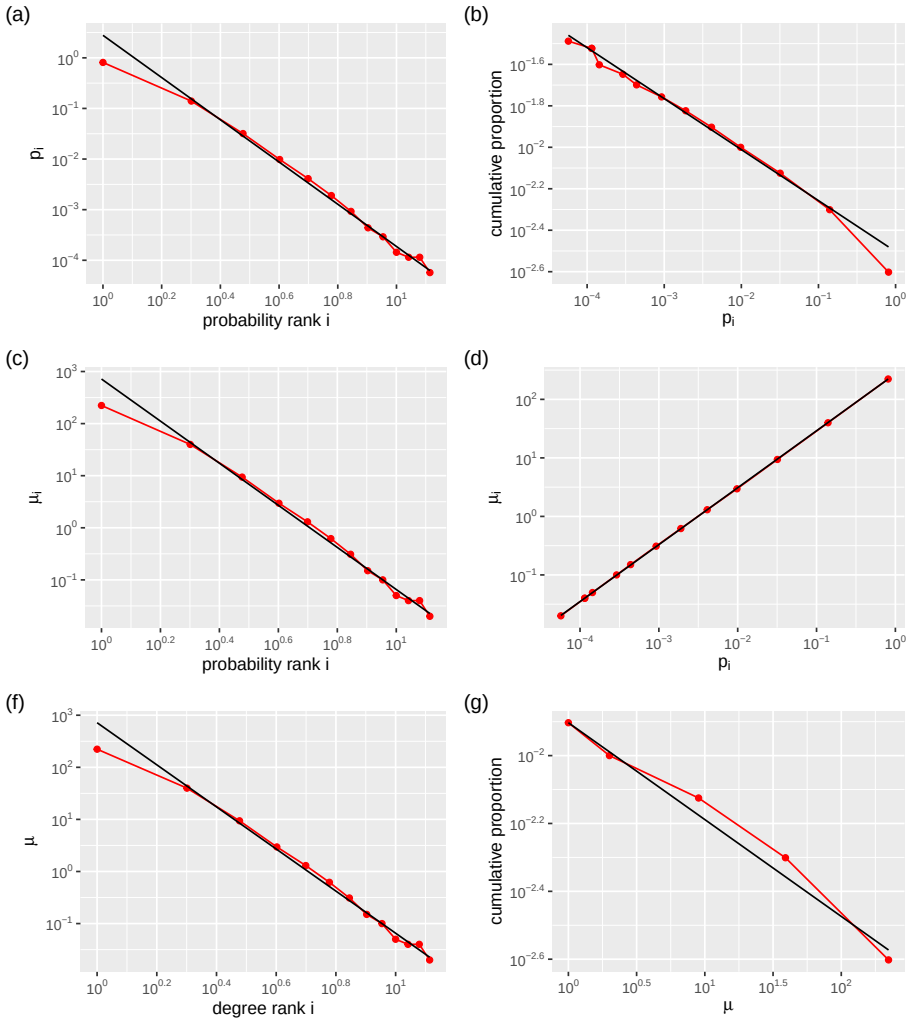


Figure 4.33: Same information as in Figure 4.18 but the model follows the equations of the external model with π following a uniform distribution and the initial condition is one to one connections between words and meanings. $\lambda^* = 0.4970$. Table 4.6 shows the values of the exponent and the factor of the fitted power law.

plot	α	k
a	2.6176507	0.6341345
b	0.3812498	0.0021506
c	1.1623482	37.4616016
d	-0.5600378	158.5629462
f	1.1623482	37.4616016
g	0.3954554	0.0250000

Table 4.5: Table showing the exponent and factor of the power laws fitted in Figure 4.32

plot	α	k
a	4.1719196	2.7845870
b	0.2458167	0.0031418
c	4.0408414	717.6995365
d	-0.9728453	272.7810669
f	4.0408414	717.6995365
g	0.2854165	0.0125000

Table 4.6: Table showing the exponent and factor of the power laws fitted in Figure 4.33

For the external model and $\phi = 1$ Figures 4.34, 4.35 and 4.36 show the information theoretic measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the random graph, single link and one-to-one initial graphs respectively. Appendix C.1 also shows the same figures but with disconnected meanings disallowed.

Figures 4.37 and 4.38 (random and one-to-one respectively) show statistical measures of select values of λ, with Figures 4.39 and 4.40 (random and one-to-one respectively) showing the fitting of the curve to a power law for a single select value of λ. While not very exact, a behavior similar to a power law can be appreciated. Tables 4.7 and 4.8 (random and one-to-one respectively) show the values of the regression exponent and factor. As with others, the single link initial condition is not plotted for select values of λ as it fails to evolve beyond the single link state. Appendix C.1 shows the same figures but with disconnected meanings disallowed.

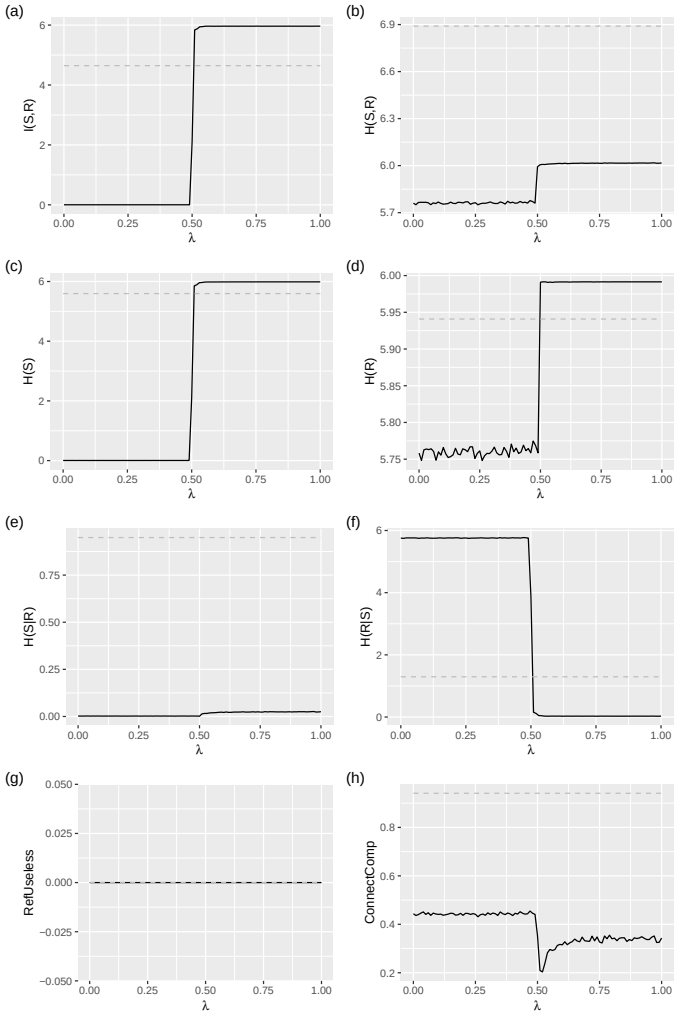


Figure 4.34: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution and with $\phi = 1$. Averages over 20 realizations.

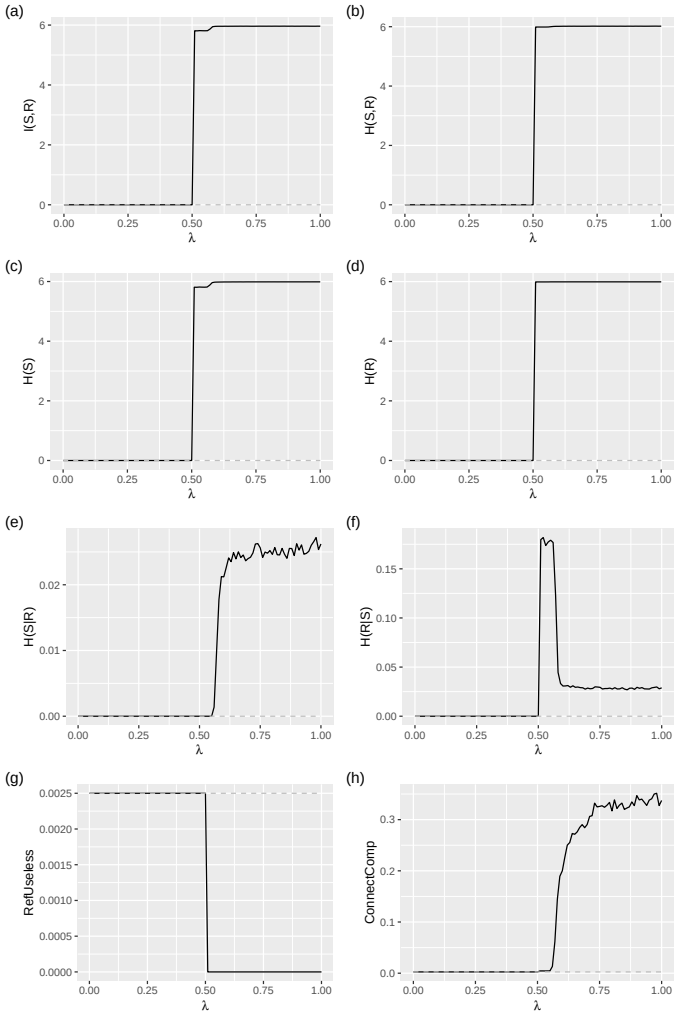


Figure 4.35: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution with $\pi = 1$. The initial condition is a single link. Averages over 20 realizations.

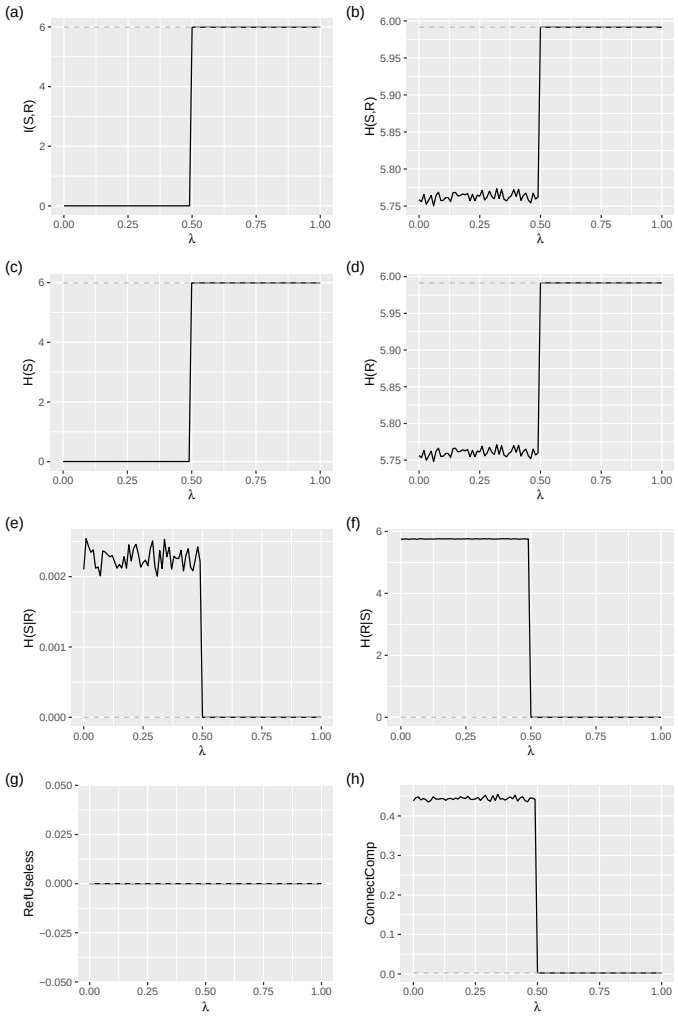


Figure 4.36: Same information as in Figure 4.13 but the graph uses the equations of the external model with π following a uniform distribution with $\pi = 1$. The initial condition is one to one connections between signals and meanings. Averages over 20 realizations.

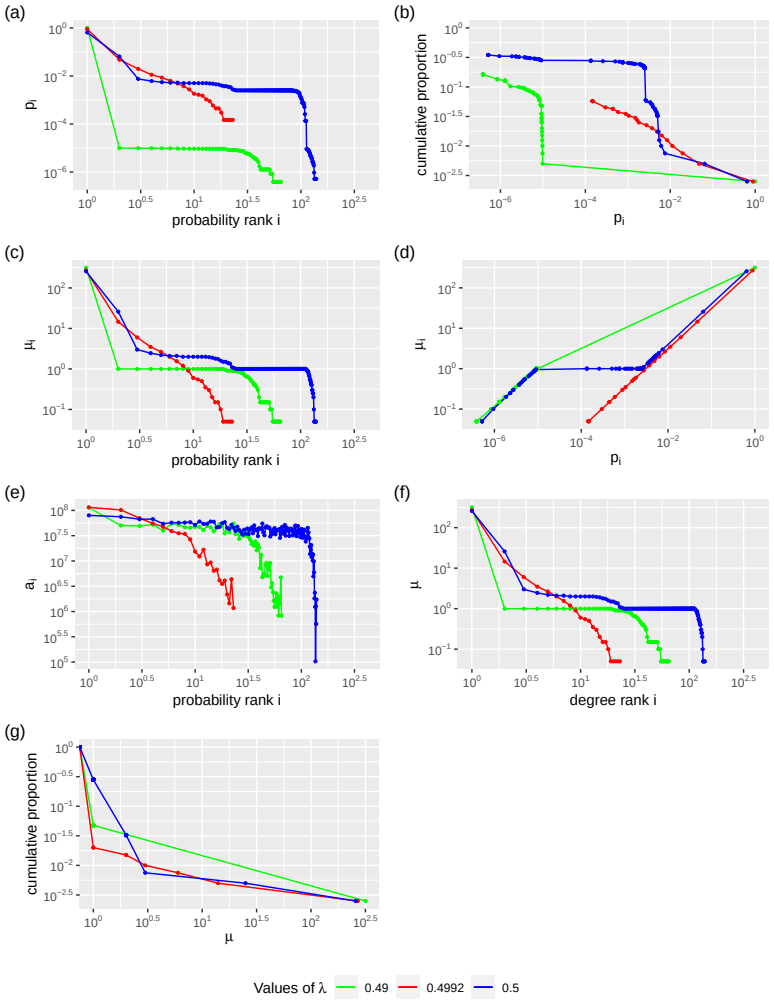


Figure 4.37: Same information as in Figure 4.16 but the graph follows the equations of the external model with $\phi = 1$ with π following a uniform distribution. $\lambda^* = 0.4992$. Averages over 20 realizations.

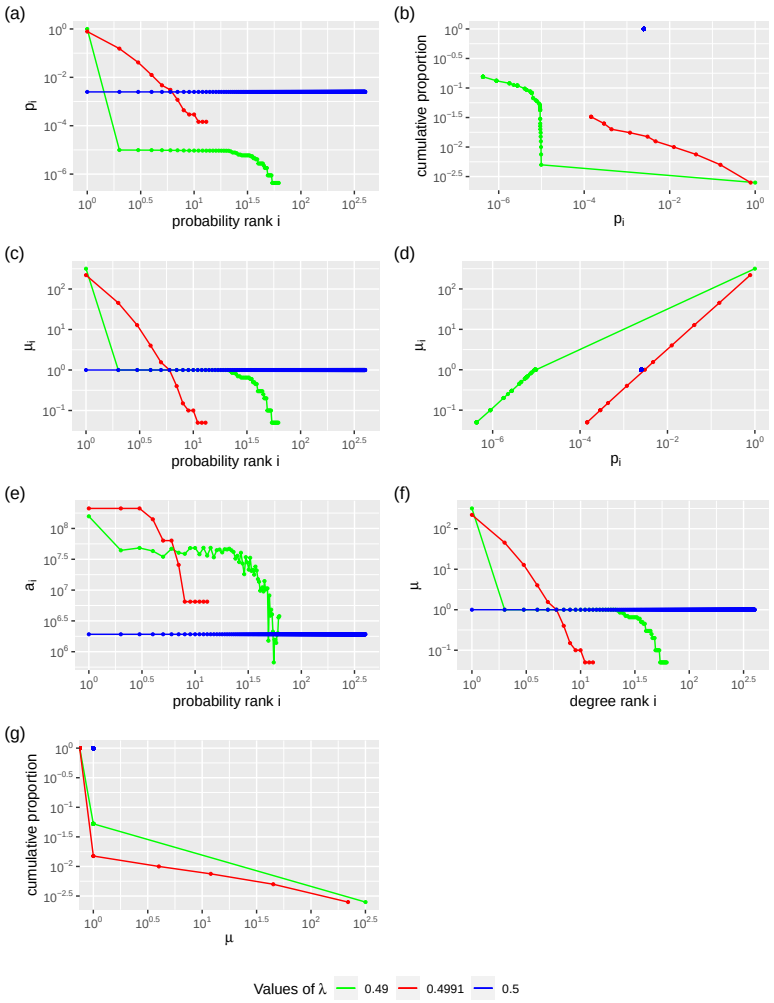


Figure 4.38: Same information as in Figure 4.16 but the graph follows the equations of the external model with π following a uniform distribution and with $\phi = 1$. The initial condition is one to one connections between words and meanings. $\lambda^* = 0.4991$. Averages over 20 realizations.

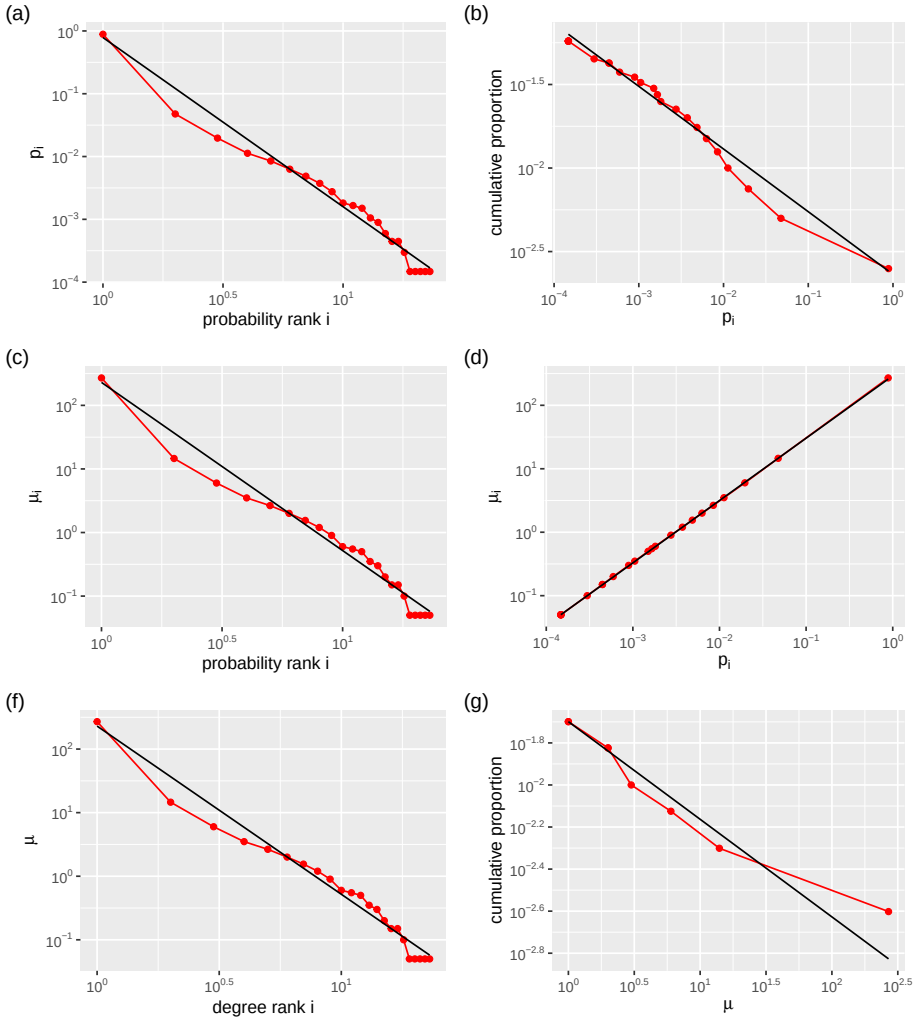


Figure 4.39: Same information as in Figure 4.18 but the model follows the equations of the external model with π following a uniform distribution and $\phi = 1$. $\lambda^* = 0.4992$. Table 4.7 shows the values of the exponent and the factor of the fitted power law.

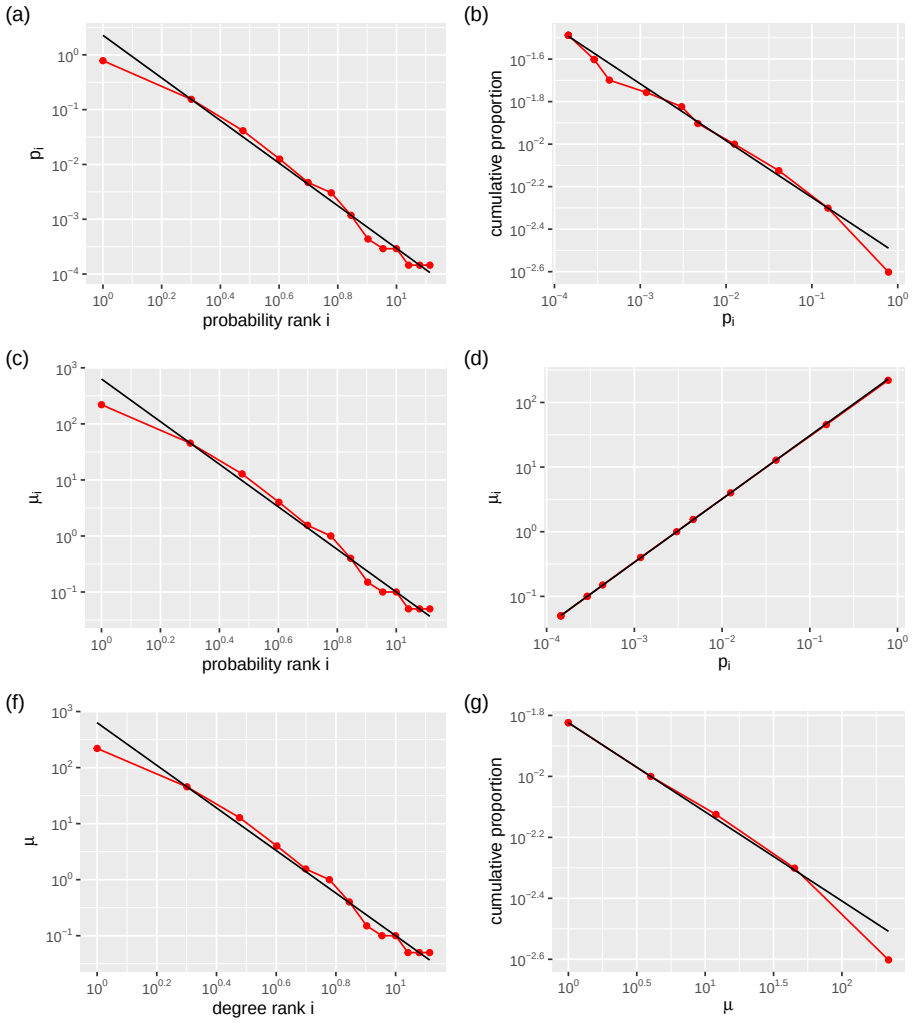


Figure 4.40: Same information as in Figure 4.18 but the model follows the equations of the external model with π following a uniform distribution and $\phi = 1$. The initial condition is one to one connections between words and meanings. $\lambda^* = 0.4991$. Table 4.8 shows the values of the exponent and the factor of the fitted power law.

plot	α	k
a	2.6942979	0.7855283
b	0.3751754	0.0023094
c	2.6454013	229.8644521
d	-0.9825933	291.4160609
f	2.6454013	229.8644521
g	0.4641743	0.0200000

Table 4.7: Table showing the exponent and factor of the power laws fitted in Figure 4.39

plot	α	k
a	3.8860954	2.27097983
b	0.2671876	0.00303420
c	3.8007579	632.05935245
d	-0.9804221	293.30509675
f	3.8007579	632.05935245
g	0.2924813	0.01500000

Table 4.8: Table showing the exponent and factor of the power laws fitted in Figure 4.40

Chapter 5

Discussion 

This chapter discusses the results obtained in Chapter 4. Section 5.1 covers the quantitative linguistics side of the results, relating them with language laws and previous research. A model should be able to make predictions. Whether this models have been able to make predictions will be discussed. Section 5.2 discusses the results relating to the computational side. It focuses specially on the optimization aspects of the model and the local minima of the cost function that may or may not be found by this approach. Finally, Section 5.3 talks about possible future work that might stem from this thesis.

5.1 Quantitative linguistics discussion

Chapter 1 already explained several models that have been used to try to explain linguistic laws, such as the random typing model or Simon's model. However, a distinction must be made between a model that simply describes a phenomenon and one that aims to explain it. There is no explanation without theory, and theory is a series of principles. Not all models can give explanations, because they cannot make predictions.

While other models can describe Zipf's law or other power laws, they cannot make predictions beyond that. One could not use Simon's model to explain how children learn new words while these models can be used to try to make this kind of prediction. [9] [4]

Here, based on the presented results, we discuss whether these models can explain certain language laws, and how strong this result actually is.

Looking at the effect of the initial conditions, in all cases the single link initial condition failed to evolve. For $\lambda \leq 0.5$ it remained a single link, while for $\lambda > 0.5$ it became a one to one bijection between words and meanings. See Figures 4.14, 4.21, 4.28 and 4.35. It can be seen how there is no intermediate stage in the phase change at $\lambda \approx 0.5$.

From a language evolution point of view, it is possible that human language follows a different cost function, one for which a single link initial condition

wouldn't fail to become a human language. However, borrowing the concepts of C.S. Peirce [1], it can be speculated that languages first developed as random iconic relationships (that is, words were similar to their meaning, for instance the utterance "bark" being similar to a dog's barking sound) and as the language evolved (optimized) symbolic relationships began to appear.

5.1.1 Zipf's word frequency laws

Zipf's law of word frequency can be observed in some form in both the internal model and the external model. Figures 4.18, 4.32 and 4.39 show that a power law appears in the frequency-rank relationship plot, subfigure (a). However, 4.25 does not show a power law. Instead a single word dominates while the rest are kept at a low frequency.

Previous results had already noted that a power law appeared for the internal and external model for $\phi = 0$ [10] [13]. And while these figures show that a power law appears for a select value of λ this does not mean that they cannot appear for others. Additionally, these figures now show that Zipf's law also appears for an initial condition other than a random bipartite graph. This makes the previously found prediction even stronger.

As for the values of the power law parameters, they are also similar to the ones found previously. In [10] a value of $\alpha \approx 1.5$ is given, which is replicated here in Table 4.1. Interestingly, when the initial condition is a one to one graph, $\alpha \approx 4$ (Table 4.2) which is a much higher value. Seeing that in real human language $\alpha \approx 1$, this does not quite predict human language. A power law is found, but the parameters are not the ones one would expect for human language.

In [13], a power law with $\alpha \approx 1$ was found for the external model when $\lambda = 0.41$. Here, values of lambda closer to 0.5 were examined and a power law was still found. However, the exponent was once again much larger than 1 for both the random bipartite graph and the one to one connections as initial conditions (Tables 4.5 and 4.6).

When $\phi = 1$ is added to the external model, $\alpha \approx 2.7$ and $\alpha \approx 3.9$ for the random and one to one initial conditions respectively. Again, far from the human language value of 1.

It's also worth noting that the curves shown follow a power law to certain degrees of similarity. Figure 4.39, for instance, is not perfectly straight in the log-log scale. While Figure 4.18 follows a line almost perfectly until the last tail of lower rank probabilities which drops drastically.

In summary, while power laws have been found, Zipf's law with a value of $\alpha = 1$ is not found.

5.1.2 Zipf's meaning frequency law

The ϕ parameter was added to the model to try and predict Zipf's meaning frequency law. [14] The relationship between the exponent of the relationship between word frequency and word frequency rank ($f \propto i^{-\alpha}$), the exponent of

the relationship between number of meanings and word frequency ($\mu \propto f^{\delta}$) and the exponent of the relationship between number of meanings and word frequency rank ($f \propto i^{-\gamma}$) should follow the relationship in Equation (1.2).

In the case of the internal model with $\phi = 0$, power laws are found in subfigures (a), (c) and (d) of Figure 4.18 (random bipartite graph as initial condition) and Figure 4.19 (one to one graph as initial condition). For the random initial condition, $\alpha \approx 1.5$, $\gamma \approx 1.5$ and $\delta \approx 1$ (Table 4.1). While the relationship from Equation (1.2) holds, it does not do so with the values of human language ($\alpha = 1$, $\gamma = \delta = 1/2$). For the one to one initial condition, $\alpha \approx 4$, $\gamma \approx 4$ and $\delta \approx 1$, the same problem as with the random initial condition but for values even farther away from human language.

When $\phi = 1$ is introduced to the first model, however, the power laws disappear. It would be expected that [14]

$$\delta = \frac{1}{\phi + 1}$$

but this is simply not the case.

As for the external model with $\phi = 0$ the relationships between degree of a word and its probability rank and also between degree of a word and its frequency do not follow power laws (Figure 4.32) for a value of λ close to 0.5. If using the approximations of a power law obtained from the regression, a value of $\delta \approx 0.5$ is recovered (Table 4.5) however the rest of parameters continue to be far from human language. When the initial graph is one to one instead of random, power laws are found (Figure 4.33) but $\delta \approx 1$.

When adding $\phi = 1$ to the external model, $\delta \approx 1$ for both initial conditions.

In summary, while the relationship from Equation (1.2) holds for many of the combinations of parameters tested, values similar to those of human language ($\delta = \gamma = 1/2$ and $\alpha = 1$) are not found. Adding the parameter $\phi = 1$ does not seem to help obtain values closer to human language. Indeed, for the internal model it removed the power law behavior.

5.1.3 Zipf's age frequency law

An interesting and somewhat unexpected result is that every single combination of parameters consistently shows Zipf's age frequency law. This is not explicitly stated as a power law and it does not appear as one in the data. However, it seems that the correlation always holds for these models: Under any combination of parameters more frequent words are older words. As seen in Chapter 1, Zipf argued that this should be the case in his book [23] and found empiric data in favor of this (Figure 1.2).

Both models very strongly reflect this law. Again, it is a consistent tendency in every result obtained, Figures 4.16, 4.17, 4.23, 4.24, 4.30, 4.31, 4.37 and 4.38.

This correlation is one of the main results of this thesis.

5.2 Computational results discussion

At the core of this model is the minimization of a cost function. This cost function is used to describe the effort of speaker and hearer of the language. For the internal model, this cost function has shown to be able to predict a bias in child vocabulary learning. [9] [4]

5.2.1 Local minima

For some combinations of parameters it is clear that a minima has been reached. This is the case when the extreme states (a single link, a one to one relationship of words and meanings) are achieved in the graph. In other cases it is not so clear that a local minima has actually been reached. This includes the cases studied in Chapter 4 with λ^* where linguistic laws could be recovered. The stronger the stop condition of the minimization algorithm, the more sure one can be that a minimum has been reached. However, this also means a longer time to obtain a result.

5.3 Future work

Much work could be further derived from the contributions presented here. Parameters could be tweaked and changed to observe various versions of the models and try to more closely obtain linguistic laws. But the more fundamental ideas presented could also be changed to improve the results and or to use different algorithms that might be more efficient and present less numerical error.

5.3.1 Optimization methods

The model is optimized by performing mutations on the boolean values of the A adjacency matrix of the graph using a Monte Carlo Markov Chain method at zero temperature.

An alternative to this is simulated annealing. That is, to use nonzero temperature for the Monte Carlo process, allowing for non optimal states to be chosen. This might help escape from states that are not local minima but that also have very few paths to other minimal states. This would only need to slightly modify the optimization algorithm to add the temperature parameter.

Another alternative optimization method is gradient descent. The cost function could be optimized as a function of A (with λ being a constant) $\Omega(A)$ would then need to be derivable. A first step to make it derivable would mean making A a matrix of reals representing weights instead of booleans representing just whether the connection exists or not. This would be a much more complex endeavor than simulated annealing. However, this would yield a method similar to what is used in AI with a much lower complexity than the models seen in AI.

5.3.2 Other values of ϕ

Other values of ϕ can be explored. Here only values of ϕ 0 or 1 are seen. Other interesting values that have not been studied in depth are 0.5 and 1.5.

This would be a very simple thing to implement, as easy as changing a configuration file (see Appendix A.2).

5.3.3 Vocabulary learning

The internal model has already been used to predict vocabulary learning biases in children, both for the case $\phi = 0$ [9] and for $\phi = 1$ [4]. A similar analysis could be done for the external model. Can it make these predictions? Under which conditions? This is purely mathematical work, without any computational aspects to it.

5.3.4 Numerical error

A great effort to find ways to reduce numerical error due to floating point arithmetic has been done as part of this thesis. However this can continue to be a problem, specially for higher values of n and m where a greater number of additions and subtractions take place. It is possible that some dynamic calculations can be done statically without sacrificing efficiency. Indeed, some variables are already calculated statically in the dynamic algorithm, as they have similar complexity in either case but involve many more additions and subtractions in the dynamic algorithm. A similar effort could be done for other sections of the dynamic computations.

Appendix A

Code

A.1 repo

- afegir repositori amb el codi

A.2 Program parameters

Here all the parameters that can be specified to the tool created as part of this thesis are explained.

- n and m, the size of the two sets in the bipartite graph
- ϕ the parameter that was added to generalize the older models
- Whether or not to use constant *a priori* probabilities for the meanings. Using them implies using the external model, not using them means using the internal model.
- Whether unlinked objects are allowed or not.
- The π parameter, the distribution of the *a priori* probabilities, can be specified in a number of ways if these probabilities are indeed being used. In addition to those specified in Section 3.1, it is allowed to manually specify an arbitrary probability distribution.
- Which λ parameters to test, the program will generate data for all the specified λ paramters. They can be specified either as a range or manually one by one.
- The number of realizations to be done. For each value of λ, that many optimization processes will be carried out and the results averaged.

- The initial graph that the optimization starts from. This can be a $G_{n,m,p}$ or $G_{n,m,e}$ random bipartite graph (see Section 3.1), a complete graph or a one to one bijection. In that last case, only the first $\min(n, m)$ words and meanings are connected.
- Whether to use the static or dynamic equations.
- The number of mutations. Either a constant number of or random binomial mutations with a given probability.
- The stop condition. Either of the weak or strong conditions, as well as the option of manually specifying a number. It is always the number of failures to improve Ω.
- A random seed can be manually specified or left random (even if random, the seed used is always logged). Whether to emit or not certain results. By default the program emits several csv files corresponding to the information theoretical plots as a function of lambda and the statistics for every value of lambda computed. It also outputs every generated graph in every realization. Any other measure could be obtained from these graphs.

Appendix B

Formulae

derivacions llargues i pagines i pagines de mates van aqui

B.1 Properties

B.1.1 Simplification of the equations of entropies

Property 1 is used in several parts of the thesis to simplify the expressions of entropies. The derivation is short, but it is not added into the main text of the thesis for the sake of clarity and focus.

$$
\begin{aligned}
&-\sum_i \frac{x_i}{T} \log \frac{x_i}{T} \\
&-\frac{1}{T} \sum_i x_i \log x_i - x_i \log T \\
&-\frac{1}{T} \sum_i x_i \log x_i + \log(T) \frac{1}{T} \sum_i x_i
\end{aligned}
$$

at this point we can apply

$$\sum_i x_i = T$$

and obtain

$$\log T - \frac{1}{T} \sum_i x_i \log x_i$$

B.1.2 Proof that $\sum_{i=1}^{n} \mu_i^{\phi} \chi_i = \rho$

In order to use Property 1 to simplify the derivation of Equation (2.44), Property 2 must hold Here it is shown that this is the case.

Starting from the equality, we can expand both sides

$$\begin{aligned}
\sum_{i=1}^{n} \mu_i^{\phi} \chi_i =& \rho \\
\sum_{i=1}^{n} \mu_i^{\phi} \sum_{j=1}^{m} \frac{a_{ij}(1-\delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}} =& \sum_{j=1}^{m}(1-\delta_{\omega_j,0})\pi(r_j) \\
\sum_{j=1}^{m} \left(\sum_{i=1}^{n} a_{ij} \mu_i^{\phi} \right) \frac{(1-\delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}} =& \sum_{j=1}^{m}(1-\delta_{\omega_j,0})\pi(r_j),
\end{aligned}$$

applying Equation (2.4) we obtain

$$\sum_{j=1}^{m}(1-\delta_{\omega_j,0})\pi(r_j)$$

on both sides.

Recall the definition of $p(s_i)$ in Equation (2.39) to see that

$$\sum_{i=1}^{n} p(s_i) = 1$$

immediately follows.

B.2 Derivation of the joint entropy in the external model

In Section 2.1.3 the derivation of the joint entropy is left as just the high level reasoning of which formulas may be used to obtain its expression. Here, the full derivation of both approaches is given.

The first approach consists of applying Equation (2.38) to the information

theory definition of $H(S,R)$ (Equation (1.7)),

$$\begin{aligned}
H(S,R) =& -\sum_{i=1}^{n}\sum_{j=1}^{m} p(s_i,r_j)\log\left[p(s_i,r_j)\right] \\
=& -\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{a_{i,j}(1-\delta_{\omega_j,0})\mu_i^{\phi}\pi(r_j)}{\rho\omega_{\phi,j}} \log\left[\frac{a_{i,j}(1-\delta_{\omega_j,0})\mu_i^{\phi}\pi(r_j)}{\rho\omega_{\phi,j}}\right] \\
=& -\frac{1}{\rho}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{a_{i,j}(1-\delta_{\omega_j,0})\mu_i^{\phi}\pi(r_j)}{\omega_{\phi,j}} \log\left[\frac{\mu_i^{\phi}\pi(r_j)}{\rho\omega_{\phi,j}}\right] \\
=& -\frac{1}{\rho}\sum_{j=1}^{m} \frac{(1-\delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}} \left[\phi\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi}\log\mu_i\right. \\
&+ \log\pi(r_j)\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi} - \log(\rho)\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi} \\
&\left. - \log(\omega_{\phi,j})\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi}\right].
\end{aligned}$$

Applying Equation (2.43) we arrive at Equation (2.42).

The second approach consists on using other less complex expressions we have available and building up from them using the definitions

$$H(S,R) = H(S|R) + H(R), \tag{B.1}$$

$$H(S|R) = \sum_{j=1}^{m} H(S|r_j)p(r_j), \tag{B.2}$$

$$H(S|r_j) = -\sum_{i=1}^{n} p(s_i|r_j)\log p(s_i|r_j). \tag{B.3}$$

Starting from Equation (B.3) and applying Equation (2.37) we obtain

$$\begin{aligned}
H(S|r_j) =& -\sum_{i=1}^{n} \frac{a_{i,j}\mu_i^{\phi}\pi(r_j)}{\rho\omega_{\phi,j}} \log\left(\frac{a_{i,j}\mu_i^{\phi}\pi(r_j)}{\rho\omega_{\phi,j}}\right) \\
=& \log\omega_{\phi,j} - \frac{1}{\omega_{\phi,j}}\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi}\log(a_{i,j}\mu_i^{\phi}) \\
=& \log\omega_{\phi,j} - \frac{\phi}{\omega_{\phi,j}}\sum_{i=1}^{n} a_{i,j}\mu_i^{\phi}\log(\mu_i)
\end{aligned}$$

and after applying Equation (2.43)

$$H(S|r_j) = \log\omega_{\phi,j} - \frac{\phi\nu_j}{\omega_{\phi,j}}. \tag{B.4}$$

Applying Equations (B.4) and (2.35) to Equation (B.2) we reach

$$\begin{aligned} H(S|R) =& \sum_{j=1}^{m} \left(\log \omega_{\phi,j} - \frac{\phi \nu_j}{\omega_{\phi,j}} \right) \frac{(1 - \delta_{w_j,0})\pi(r_j)}{\rho} \\ =& \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{w_j,0})\pi(r_j) \left[\log(\omega_{\phi,j}) - \frac{\phi \nu_j}{\omega_{\phi,j}} \right]. \end{aligned} \tag{B.5}$$

The last step consists of applying Equations (B.5) and (2.41) to Equation (B.1), obtaining Equation (2.42) again

$$\begin{aligned} H(S,R) =& \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{w_j,0})\pi(r_j) \left[\log(\omega_{\phi,j}) - \frac{\phi \nu_j}{\omega_{\phi,j}} \right] \\ &+ \log \rho - \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{w_j,0})\pi(r_j) \log \pi(r_j) \\ =& \log \rho - \frac{1}{\rho} \sum_{j=1}^{m} (1 - \delta_{w_j,0})\pi(r_j) \left[\frac{\phi \nu_j}{\omega_{\phi,j}} + \log \frac{\pi(r_j)}{\omega_{\phi,j}} \right]. \end{aligned}$$

B.3 Dynamic Equations for the external model

The full derivation of the dynamic equations for the external model is given in this section. The equations themselves as well as other definitions used here are already given in Section 2.1.3.

In this section, the derivation is logical rather than mathematical, consisting of explanations behind the logic of the equations.

ρ (see Equation (2.36)) only changes when, as a result of the mutation, the meaning r_j either was connected and becomes disconnected or was disconnected and becomes connected, resulting in Equation (2.55).

ν_l is very similar to $\omega_{\phi,l}$ (compare Equations (2.43) and (2.4)). Equation (2.56) shows how ν_l changes in the same cases and in the same way as $\omega_{\phi,l}$ (Equation (2.29)).

χ_k depends on $\omega_{\phi,l}$ such that $l \in \Gamma_S(k)$ (see Equation (2.40)). For χ_i, the entire value has to be recalculated, as every $\omega_{\phi,l}$ for $l \in \Gamma_S(k)$ will have changed (Equation (2.29)). It is more efficient (and reduces the amount of floating point error) to calculate χ_i statically than to subtract every $\omega_{\phi,l}$ and then add every $\omega'_{\phi,l}$. For all other values of χ_k ($k \neq i$), the affected $\omega_{\phi,l}$ are the intersection of the set of neighbors of i plus the meaning r_j and the set of neighbors of k. This is the set $A_{i,j}(k)$ defined in Equation (2.53). With this reasoning we reach the formula for the dynamic recalculation of χ_k, Equation (2.57).

$X(R)$ (Equation (2.50)) changes in a very similar way to ρ (Equation (2.36)), as they both depend on the same variable in the same way: only when the meaning r_j becomes connected or disconnected does $X(R)$ change, resulting in Equation (2.55).

$X(S,R)$ (Equation (2.48)) will change when a $x(r_l)$ (Equation (2.52)) changes. Both $\omega_{\phi,l}$ and ν_l change for $l \in \Gamma_S(i) \cup \{j\}$ (Equations (2.29) and (2.56)). So these are the components $x(s_l)$ of $H(S,R)$ that will change. For the special case of $l = j$ we should not subtract the old value when r_j has become connected as a result of the mutation (as that component was not present), and we should not add the new value when r_j has become disconnected (as that component should not be present). Or in other words, only subtract the old value if r_j was connected before the mutation (this serves to either update it or to remove it) and only add the new value if r_j is connected after the mutation (this serves to either update it or to add it). The resulting expression for $X'(S,R)$ is in Equation (2.59).

$X(S)$ (Equation (2.49)) will change when $x(s_k)$ (Equation (2.51)) changes, which depends on χ_k (Equation (2.57)). χ_k is updated for every k such that $|A_k| \neq \emptyset$ and χ_i is always updated, which is the definition of the $B_{i,j}(k)$ set (Equation (2.54)). Using $B_{i,j}(k)$ we obtain the expression for $X'(S)$ in Equation (2.60).

Appendix C

Figures

figures que sobren

C.1 Figures of disconnected meanings for the external model

Section 4.2.2 shows the results obtained for the current model. For comparison with [13], figures generated from the same sets of parameters as for the external model but with disconnected meanings kept disallowed are shown here. The single connection initial state is never shown, as it is an invalid configuration.

For the case $\phi = 0$, Figures C.1 and C.2 show the information theoretical measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the initial graph being a random bipartite graph and a one to one configuration respectively.

Figures C.3 and C.4 (random and one to one respectively) show statistical measures for select values of λ, with figures C.5 and C.6 showing the fitting of the curve to a power law for a select value of λ. None of the studied values of λ between 0.49 and 0.5 resulted in a power law. Tables C.1 and C.2 show the values of the regression's exponent and factor

plot	α	k
a	5.1239660	1744404.5869742
b	0.1949778	0.0413557
c	1.5237337	1164.4863367
d	-0.5072076	52.8958272
f	1.5439705	1287.7147444
g	0.8441915	0.2950000

Table C.1: Table showing the exponent and factor of the power laws fitted in Figure C.5

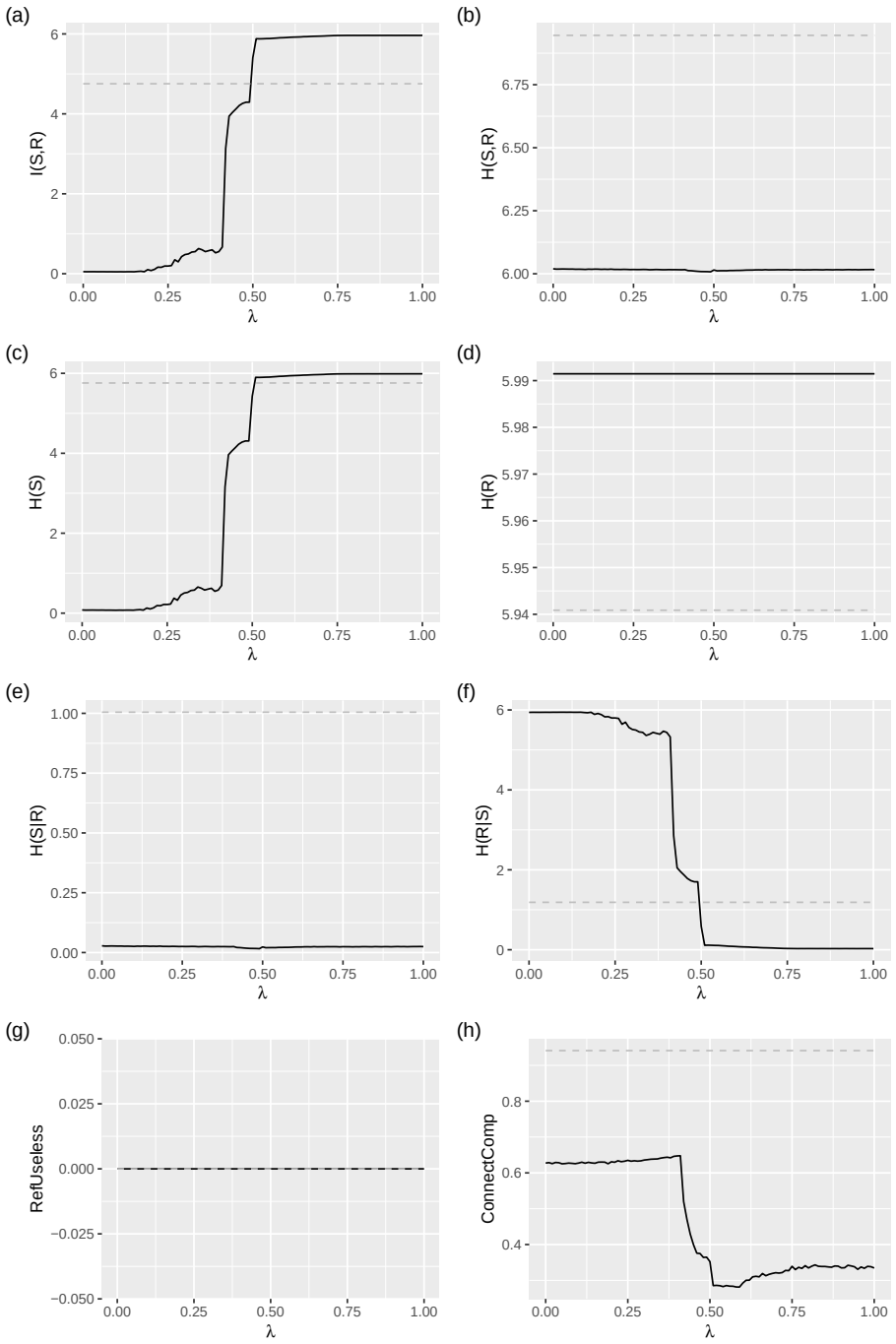


Figure C.1: Same information as in Figure 4.27 but disconnected meanings are disallowed.

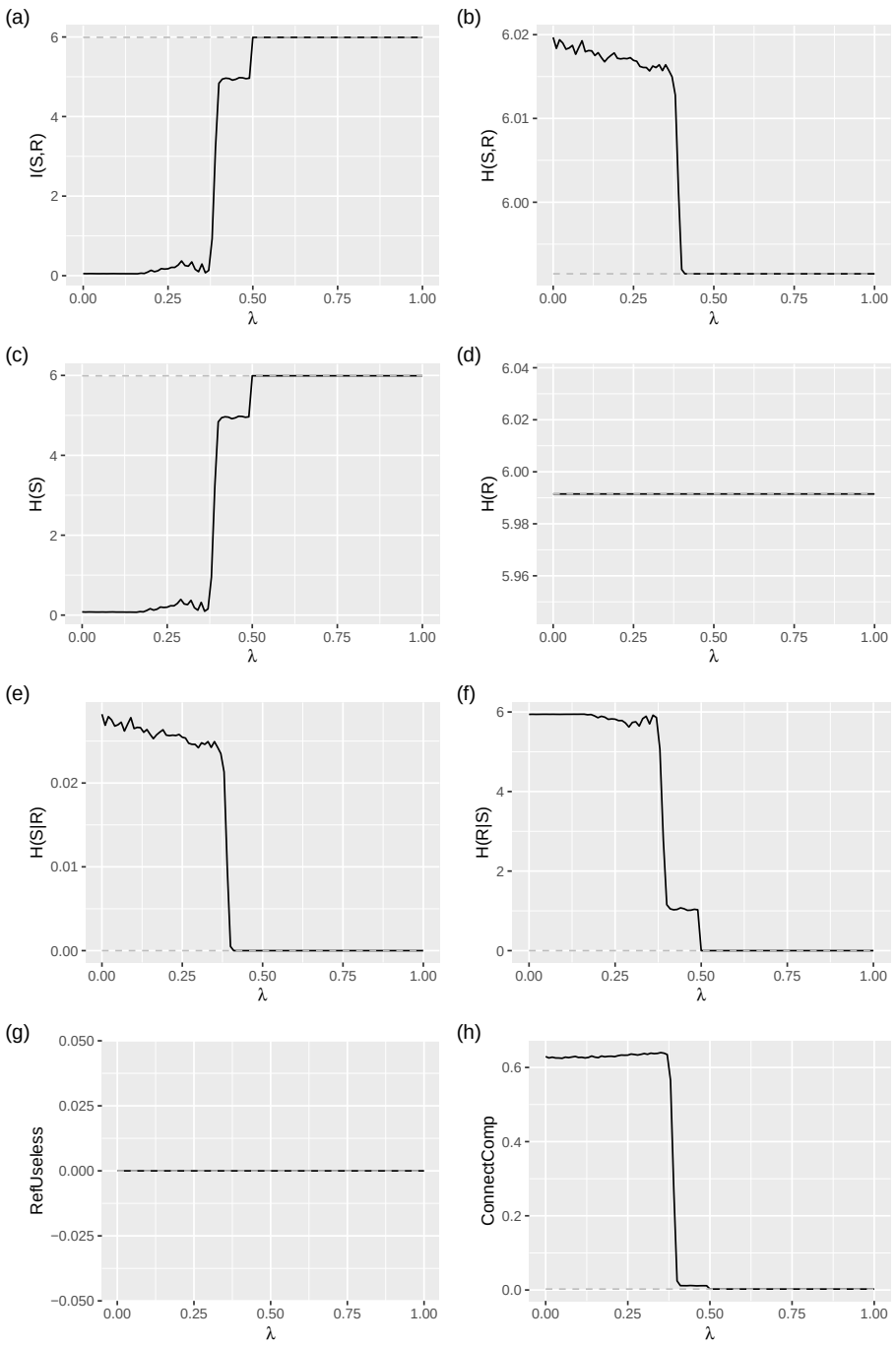


Figure C.2: Same information as in Figure 4.29 but disconnected meanings are disallowed.

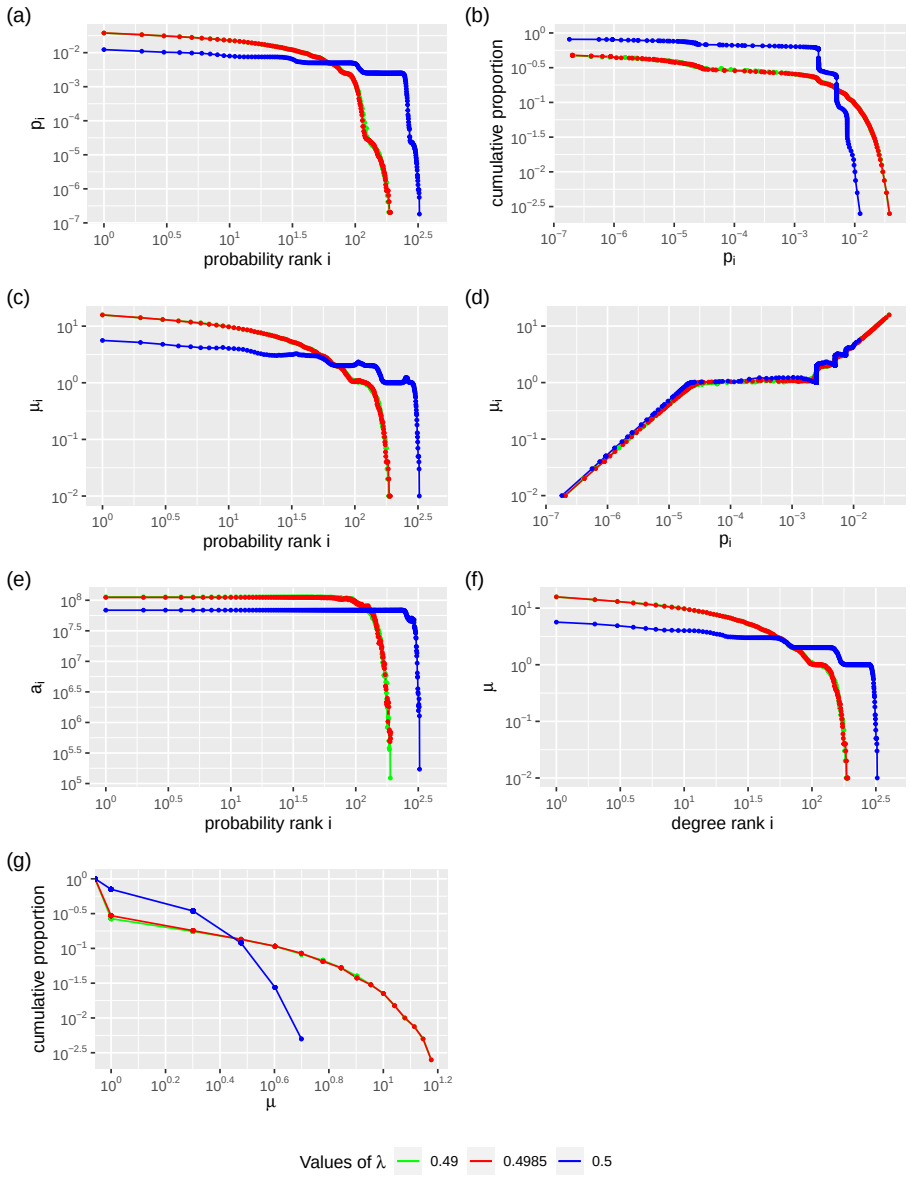


Figure C.3: Same information as in Figure 4.30 but disconnected meanings are disallowed.

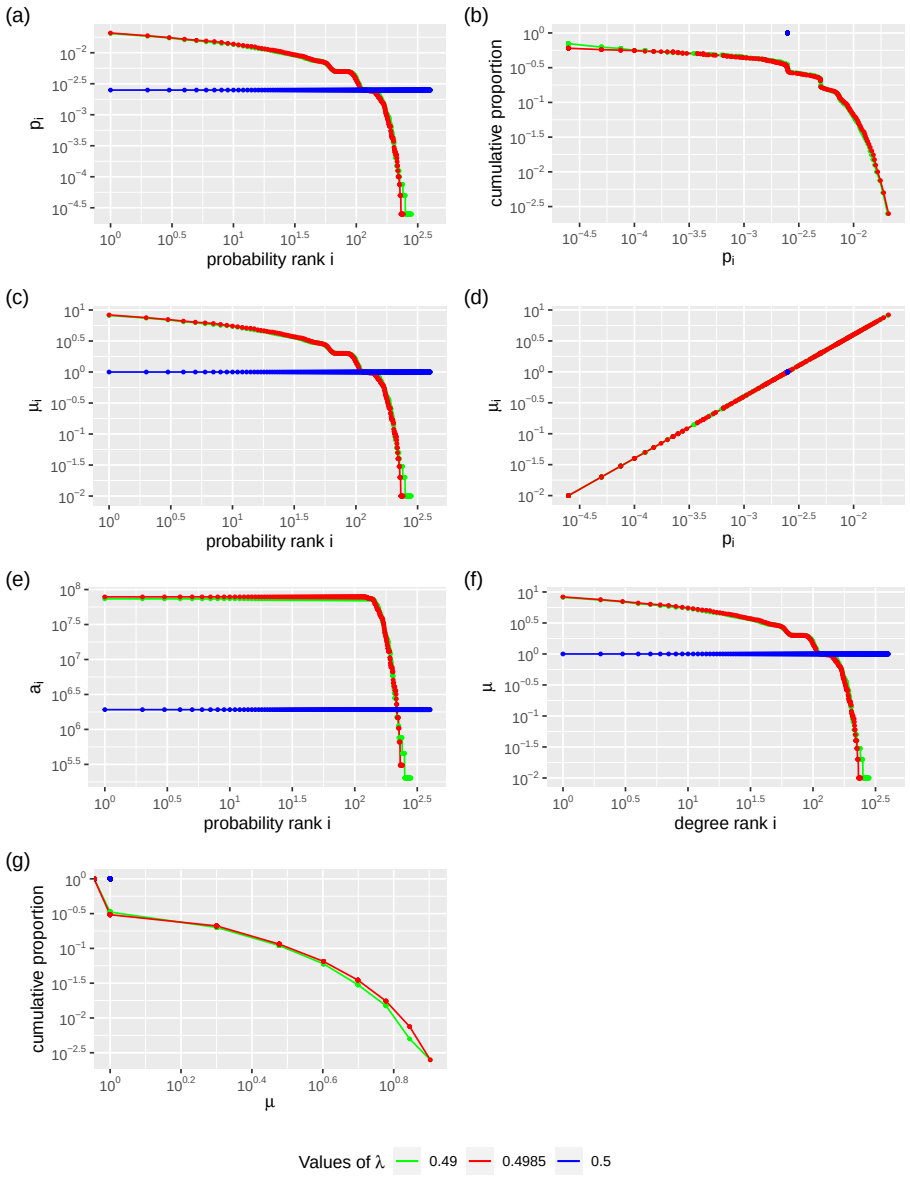


Figure C.4: Same information as in Figure 4.31 but disconnected meanings are disallowed.

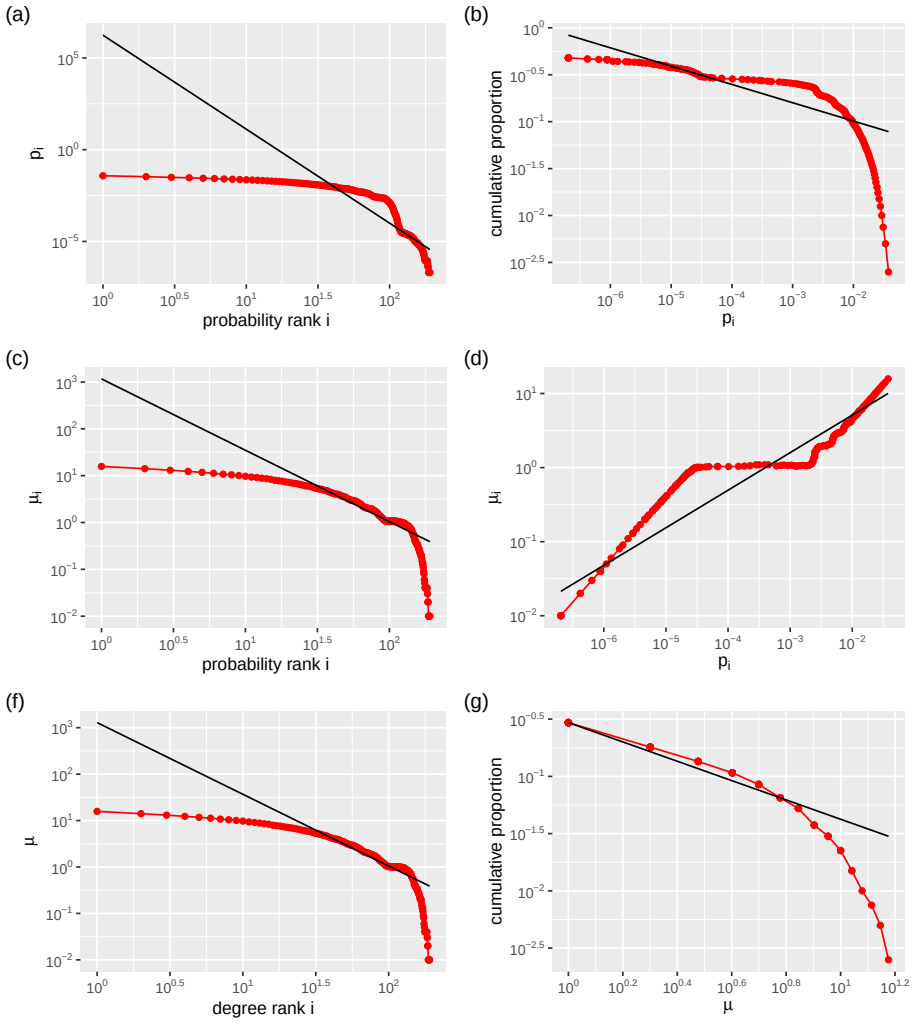


Figure C.5: Same information as in Figure 4.32 but disconnected meanings are disallowed. Table C.1 shows the values of the exponent and the factor of the fitted power law.

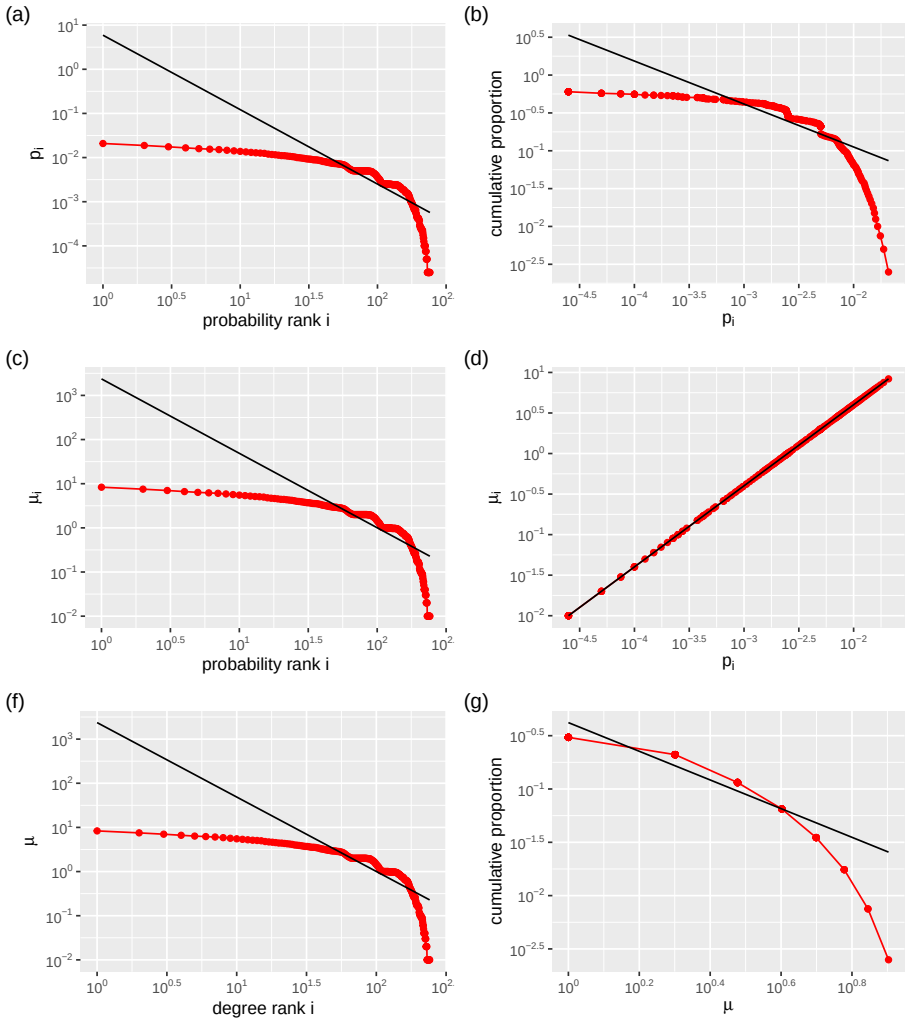


Figure C.6: Same information as in Figure 4.33 but disconnected meanings are disallowed. Table C.2 shows the values of the exponent and the factor of the fitted power law.

plot	α	k
a	1.6856845	5.9091866
b	0.5677904	0.0082092
c	1.6856845	2363.6746229
d	-1.0000000	400.0000000
f	1.6856845	2363.6746229
g	1.3451676	0.4195512

Table C.2: Table showing the exponent and factor of the power laws fitted in Figure C.6

For the case $\phi = 1$, Figures C.7 and C.8 show the information theoretical measures of the optimal graph for values of λ ranging from 0 to 1. They correspond to the initial graph being a random bipartite graph and a one to one configuration respectively. It is interesting to see that when $\phi = 1$ the one to one configuration fails to evolve for any value of λ.

Figure C.3 shows statistical measures for select values of λ, with figure C.5 showing the fitting of the curve to a power law for a select value of λ. Table C.1 shows the values of the regression's exponent and factor

plot	α	k
a	2.5318774	0.5389468
b	0.3865160	0.0022479
c	0.4151170	4.7225143
d	-0.4419160	166.3791217
f	0.4151170	4.7225143
g	2.1132828	0.2250000

Table C.3: captionTable showing the exponent and factor of the power laws fitted in Figure C.10

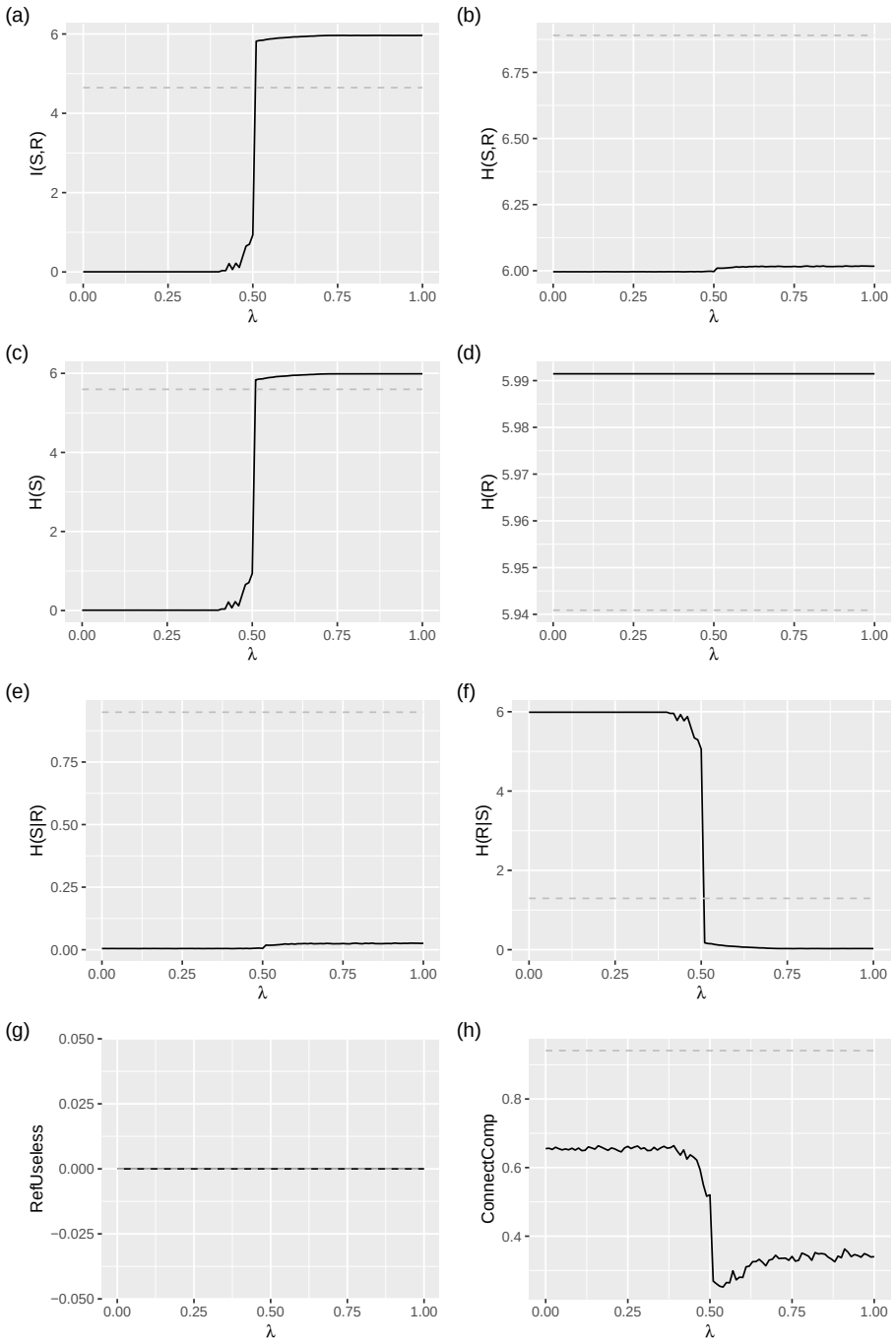


Figure C.7: Same information as in Figure 4.34 but disconnected meanings are disallowed.

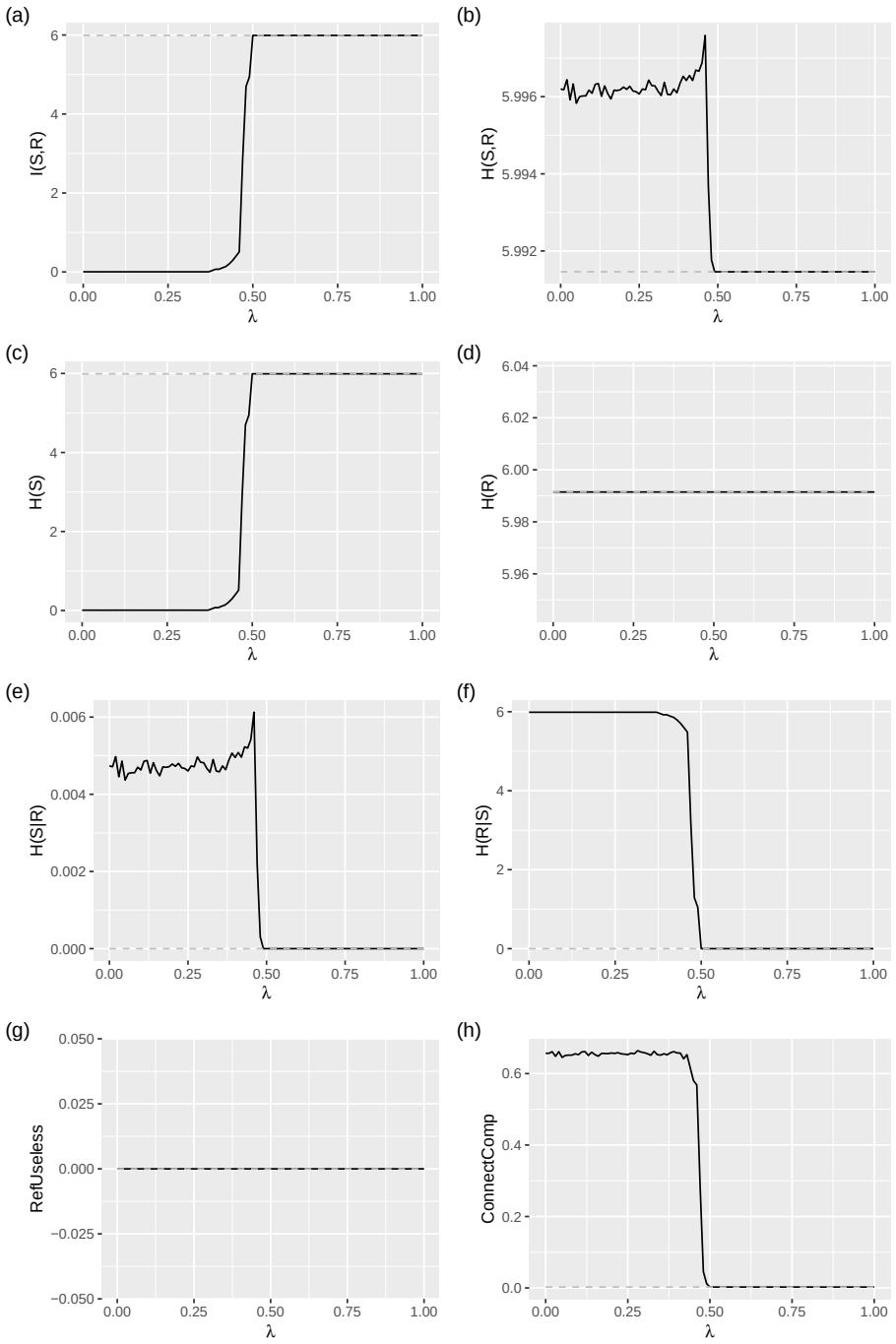


Figure C.8: Same information as in Figure 4.36 but disconnected meanings are disallowed.

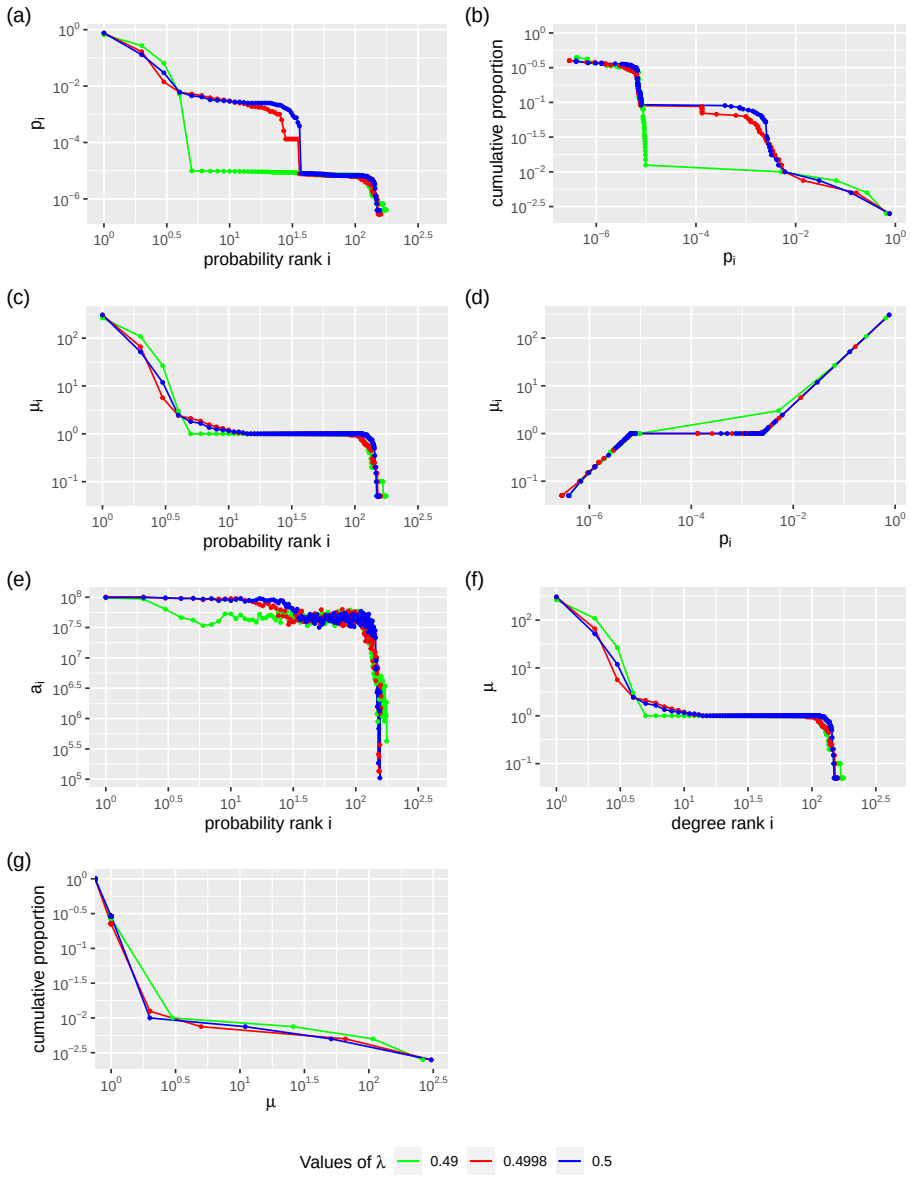


Figure C.9: Same information as in Figure 4.37 but disconnected meanings are disallowed.

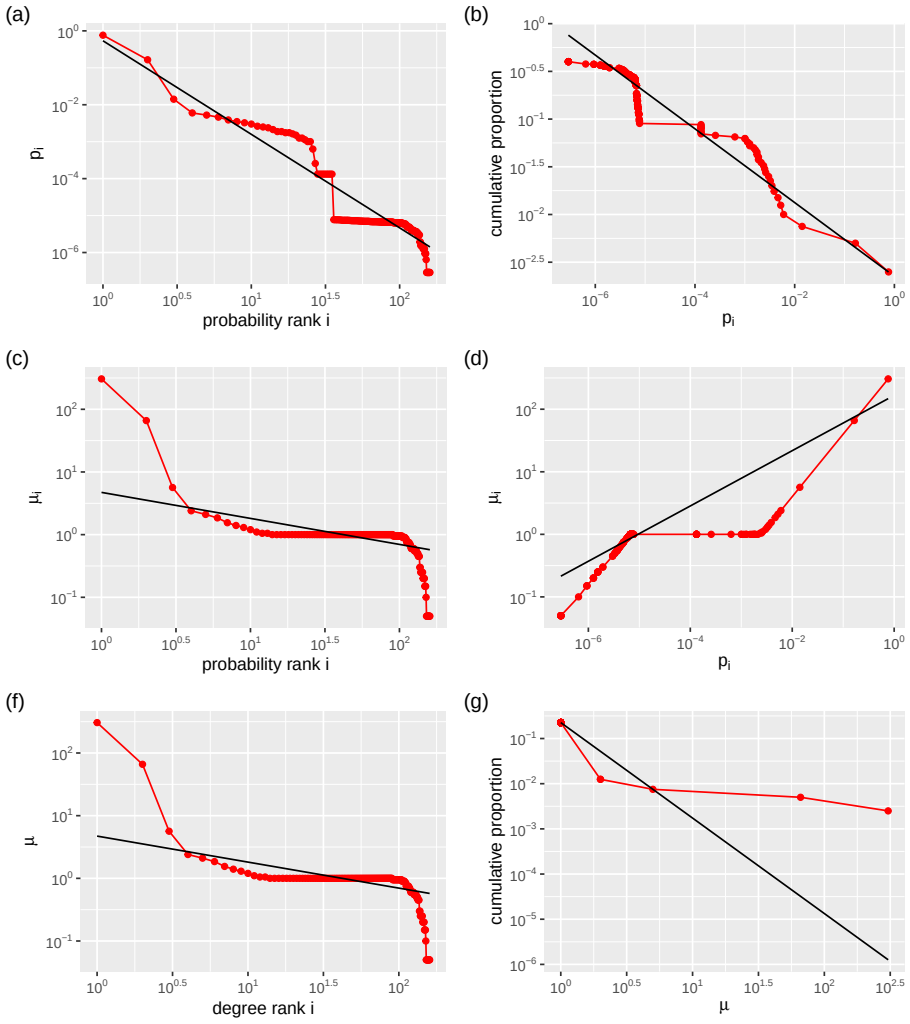


Figure C.10: Same information as in Figure 4.39 but disconnected meanings are disallowed. Table C.3 shows the values of the exponent and the factor of the fitted power law.

Bibliography

[1] Albert Atkin. "Icon, index, and symbol". In: *The Cambridge Encyclopaedia of the Language Sciences*. Ed. by Patrick Colm Hogan. Cambridge: Cambridge University Press, 2010, pp. 367–368. ISBN: 9780521866897.

[2] Andrea Baronchelli et al. "Sharp transition towards shared vocabularies in multi-agent systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2006.06 (2006), P06014.

[3] Luciana S Buriol, Mauricio GC Resende, and Mikkel Thorup. "Speeding up dynamic shortest path algorithms". In: *Rapport technique TD-5RJ8B, AT\&T Labs Research* (2003), p. 41.

[4] David Carrera-Casado and Ramon Ferrer-i-Cancho. "The advent and fall of a vocabulary learning bias from communicative efficiency". In: *Biosemiotics* (2021). URL: http://arxiv.org/abs/2105.11519.

[5] Noam Chomsky and George Miller. "Finitary models of language users". In: *Handbook of mathematical psychology* 2 (1963), pp. 419–491.

[6] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[7] Paul Erdös and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math* 5.1 (1960), pp. 17–60.

[8] Ramon Ferrer-i-Cancho. "Compression and the origins of Zipf's law for word frequencies". In: *Complexity* 21 (2016), pp. 409–411. DOI: 10.1002/cplx.21820.

[9] Ramon Ferrer-i-Cancho. "The optimality of attaching unlinked labels to unlinked meanings". In: *Glottometrics* 36 (2017), pp. 1–16.

[10] Ramon Ferrer-i-Cancho. "Zipf's law from a communicative phase transition". In: *European Physical Journal B* 47.3 (2005), pp. 449–457. DOI: 10.1140/epjb/e2005-00340-y.

[11] Ramon Ferrer-i-Cancho, Christian Bentz, and Caio Seguin. "Optimal Coding and the Origins of Zipfian Laws". In: *Journal of Quantitative Linguistics* (2020). DOI: 10.1080/09296174.2020.1778387.

[12] Ramon Ferrer-i-Cancho and Albert Díaz-Guilera. "The global minima of the communicative energy of natural communication systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 06009.6 (2007). DOI: 10.1088/1742-5468/2007/06/P06009.

[13] Ramon Ferrer-i-Cancho and Ricard V. Sole. "Least effort and the origins of scaling in human language". In: *Proceedings of the National Academy of Sciences of the United States of America* 100.3 (2003), pp. 788–791. DOI: 10.1073/pnas.0335980100.

[14] Ramon Ferrer-i-Cancho and Michael S. Vitevitch. "The origins of Zipf's meaning-frequency law". In: *Journal of the Association for Information Science and Technology* 69.11 (Nov. 2018), pp. 1369–1379. DOI: 10.1002/asi.24057.

[15] Ali Mehri and Maryam Jamaati. "Variation of Zipf's exponent in one hundred live languages: A study of the Holy Bible translations". In: *Physics Letters A* 381.31 (2017), pp. 2470–2477. DOI: 10.1016/j.physleta.2017.05.061.

[16] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2017.

[17] Mikhail Prokopenko et al. "Phase transitions in least-effort communications". In: *Journal of Statistical Mechanics: Theory and Experiment* 2010.11 (2010). DOI: 10.1088/1742-5468/2010/11/P11025.

[18] Terry Regier, Paul Kay, and Naveen Khetarpal. "Color naming reflects optimal partitions of color space". In: *Proceedings of the National Academy of Sciences of the United States of America* 104.4 (2007), pp. 1436–1441. DOI: 10.1073/pnas.0610341104.

[19] Christoph Salge et al. "Zipf's law: Balancing signal usage cost and communication efficiency". In: *PLoS ONE* 10.10 (2015), pp. 1–14. DOI: 10.1371/journal.pone.0139475.

[20] Sergio Jimenez. *Zipf's law*. [Online; accessed September 26, 2021]. 2015. URL: https://commons.wikimedia.org/wiki/File:Zipf_30wiki_en_labels.png.

[21] Herbert A Simon. "On a Class of Skew Distribution Functions". In: *Biometrika* 42.3/4 (1955), p. 425. DOI: 10.2307/2333389.

[22] J. S. Smart. "Statistical tests of the broken-stick model of species-abundance relations". In: *Journal of Theoretical Biology* 59.1 (1976), pp. 127–139. DOI: 10.1016/S0022-5193(76)80027-6.

[23] G. K. Zipf. *Human behaviour and the principle of least effort*. Cambridge (MA), USA: Addison-Wesley, 1949.