# Skim-reading thousands of documents in one minute: Data indexing and visualization for multifarious search

Alessandro Perina
Microsoft Research and
Istituto Italiano di Tecnologia
Redmond WA / Genova Italy

Dongwoo Kim
KAIST
Daejeon, Korea

Andrzej Turski
Microsoft Corporation
Redmond, WA

Nebojsa Jojic*
Microsoft Research
Redmond, WA
*Corresponding author:
jojic@microsoft.com

## ABSTRACT

In this paper we present an interface based on a recent generative model, the counting grid, here re-introduced in its basic version and largely revised to allow it to deal with large corpora. We show that it is possible to visualize thousands of high order word co-occurrence patterns by only viewing for a few minutes a new embedding we propose for text visualization, browsing and search purposes. We performed preliminary experiments with user tasks such as word spotting, rapid content search and collateral information acquisition.

## 1. INTRODUCTION

Embedding text documents into a 2D space (e.g. [13, 3]) has always been an appealing idea: If we can turn a discrete complex dataset into something that looks like an image, perhaps our brains' low to medium-level processing layers will take the lead and help us consume the dataset in a flash, the way our eyes process almost any natural image. The old idea that various types of knowledge may already be captured in image-like mental representations in our mind [8] further strengthens our expectation that even the knowledge that is inherently as discrete, hierarchical and propositional as that encoded using language, can be transformed into something continuous and referentially isomorphic, a data-driven smooth mapping that our eyes can easily saccade over. Another vehicle for of obtaining a "birds-eye-view" is the notion of the word/tag cloud where a smaller or larger handful of characteristic words is shown to the user as a summary and a very rudimentary index of the data.

However, multiple dangers lurk here. Our eyes saccade over text differently than over natural images [10, 7, 2]. The speed of visual word recognition is highly dependent on the words' immediate context, which can both speed it up *and* slow it down [2].
This of course has consequences to visualization and user interface design. For example, a 2D embedding of titles in a distance-based document embedding is hard to make sense of as the processing required for us to understand the discovered links is at a too high a level to gel well with the visual traversing paradigm. High level category labels are often added to aid the user in making sense of different areas in the embedding, but as indicated above, these labels are likely to make it even more difficult to understand the outliers that happen around the boundaries. This is why some visualizations only show documents as dots of different colors indicating broad categories, but essentially hiding all of their content until the user mouses over. Data that way does become more image-like but is akin to a very simple image.

On the other hand showing a large number of constitutive words from a document is problematic due to the users' reading habits. For instance, alphabetically arranged tags can easily be misinterpreted by a user who tends to look for a meaning in groups of words, and so the sequence of tags "living man missing money news" from a word cloud from one day of CNN news may all refer to different news stories, yet it is difficult for a human reader not to jump to a conclusion that either money or the man is missing.

It has been shown, e.g. in [12], that semantic organization of words significantly affects the user' interaction with the data, making lower-level connections (folksonomy based) better suited for consumption than the higher level language models. Thus it is not surprising that most previous user studies of various text visualization techniques similar to these resulted in the conclusion that when the user is interested in a very specific bit of information, the regular search engine interface will suffice, and that in most other situations the beneficial effects of the visualization are hard to quantify, other than through user satisfaction levels. Users tend to favor these tools, perhaps because, as we stated above, the idea of being able to extract the essence of the data and lay it out onto the screen in a rich, yet easy to grasp manner is just as appealing to the users as it is to the researchers, even if it is hard to realize.

In this paper we present an interface built upon the recent Counting Grid model [6] and we strongly believe that the approach may be a step forward. We also propose few learning algorithm aimed at avoiding local minima and producing more grids for usable for users.

### 1.1 The counting grid: A way forward?

We imagine a large grid of cells, each with a few words of different weights so chosen so that words collected from any single document in the dataset can be represented well by the weighted words in one small window encompassing several cells in the grid Fig. 1a. Aided with a good optimization technique and a user interface that fits the model well, several very interesting properties of such an embedding arise.

Firstly, it is possible to make the mapping very dense, avoiding the excessive levels of empty space in typical distance-preserving

embedding methods (note that our visual system distorts distances, see, for example [5]).

Secondly, in such dense mappings the grid is too small to avoid overlaps of windows, and so then the extent of the similarity of the nearby documents in terms of simple word usage statistics can readily be seen directly in the grid: The words shared between the two documents will tend to be seen in the region of the overlap of the two windows. Thirdly, if we travel slowly across the grid and look at the documents mapped there, we should often see gradual thematic shifts as the words early in our path are dropped and new ones are added, but the overlap in content between our new area of focus and the one just before tends to stay high. Obviously for diverse enough datasets, occasionally the smoothness in theme shifts will have to be violated in areas where two different topics expanding from different points clash in a single area creating a rift between two less related groups of documents. Finally, in most places we look, the words we can get from the nearby cells will tend to be highly related, and this should make it easier to perform visual word recognition tasks if all these words are shown on the screen, such as word spotting in a search for a particular word it should be often easy to pick out document groupings, focus on one of the relevant ones, and then follow the trail to the point of interest, then jump to another grouping of interest and focus on the new area, etc.

We call this model the counting grid, as it is a grid of word counts, and in the next section we state this idea mathematically. Then, we describe the techniques needed to properly optimize and present the counting grid to the user as an interface to various medium-sized datasets (cooking recipes, research papers, movie descriptions, etc.). Finally, we demonstrate that our interface does indeed expose high order statistics (word co-occurrence statistics beyond pairs) which then become a powerful visualization tool for both understanding the extent of the dataset and discovery of items of interest. We show that both the word combinations are meaningful beyond what was previously attempted, increasing the word spotting speeds, and that they lead to good indexing of a diverse dataset enabling users to perform dozens of semi-related search tasks in parallel in mere minutes and then walk away with much more collateral information that seeped into their brains serendipitously.

**Algorithm 1:** EM-Algorithm to learn the Counting Grids.

**Input**: Bag of words, $c_z^t$ for each sample
**while** *Convergence* **do**
  % E-Step ;
  **foreach** *Sample* $t = 1 \dots T$ **do**
    1. Update $q_{\mathbf{k}}^t \propto \exp \sum_z c_z^t \cdot \log h_{\mathbf{k},z}$ ;
  % M-Step ;
  2. Update $\pi_{\mathbf{k},z} \propto \pi_{\mathbf{k},z}^{old} \cdot \sum_t c_z^t \sum_{\mathbf{i}|\mathbf{k}\in W_{\mathbf{i}}} \frac{q_{\mathbf{i}}^t}{h_{\mathbf{i},z}}$ ;
  3. Compute $h_{\mathbf{k},z} = \frac{1}{W_1 \times W_2} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$;
  4. Compute the Log-Likelihood (Eq. 1) ;
  5. Check for convergence ;
6. Return $\pi_{\mathbf{k},z}$ and $\{q_{\mathbf{k}}^t\}$ ;

## 2. THE COUNTING GRID MODEL

The counting grid consists of a set of discrete locations in a map of arbitrary dimensions ($32 \times 32$ or $64 \times 64$ in the examples used in this paper). Each location contains a different set of weights for the each of the words in the vocabulary. A document has its own word usage counts $c_z$ and the assumption of the counting grid model is that this word usage pattern is well represented at some location $i$ in the grid. The window floating over the grid captures well variation in certain types of documents where we can see slow evolution of the topics, where certain words are dropped and new ones introduced.

A particular example of a counting grid and its weights are illustrated in Fig. 1 using font size variation, but showing only the top 3 words at each location. The shaded cells are characterized by the presence, with a non-zero probability, of the word "bake"[1]. On the grid we also show the windows $\mathbf{W}$ for 5 recipes. *Nomi* (1), an Afghan egg-based bread, is close to the recipe of the usual *pugliese bread* (2), as indeed they share most of the ingredients and procedure. Note how moving from (1) to (2) the word "egg" is dropped. Moving to the right we encounter the *basic pizza* (3) whose dough is very similar to the bread's. Continuing to the right words often associated to desserts like sugar, almond, etc emerge. It is not surprising that baked desserts such as *cookies* (4), and pastry in general, are mapped here. Finally further up we encounter other desserts which do not require baking, like *tiramisu* (5), or *chocolate crepes*.

Formally, the basic counting grid $\pi_{\mathbf{i},z}$ is a set of normalized counts of words / features indexed by $z$ on the 2-dimensional discrete grid indexed by $\mathbf{i} = (i_1, i_2)$ where each $i_d \in [1 \dots E_d]$ and $\mathbf{E} = [E_1, E_2]$ describes the extent of the counting grid. Since $\pi$ is a grid of distributions, $\sum_z \pi_{\mathbf{i},z} = 1$ everywhere on the grid. A given bag of words/features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found somewhere in the counting grid. In particular, using windows of dimensions $\mathbf{W} = [W_1, W_2]$, each bag can be generated by first averaging all counts in the window $W_{\mathbf{k}} = [\mathbf{k}, \dots, \mathbf{k} + \mathbf{W}]$ starting at grid location $\mathbf{k}$ and extending in each direction by $W_d$ grid positions to form the histogram $h_{\mathbf{k},z} = \frac{1}{W_1 \times W_2} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$, and then generating a set of features in the bag. In other words, the position of the window $\mathbf{k}$ in the grid is a latent variable given which the probability of the bag of features $\{c_z\}$ is

$$p(\{c_z\}|\mathbf{k}) = \prod_z (h_{\mathbf{k},z})^{c_z} = \frac{1}{W_1 \times W_2} \prod_z (\sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z})^{c_z},$$

Fine variation achievable by moving the windows in between any two close by but non-overlapping windows is useful if we expect such smooth thematic shifts to occur in the data, and we illustrate in our experiments that indeed it does.

To learn a Counting Grid we need to maximize the likelihood of the data:

$$\log P = \sum_t \log \Big( \sum_{\mathbf{k}} \cdot \prod_z (h_{\mathbf{k},z}^{c_z^t}) \Big) \quad (1)$$

The sum over the latent variables $\mathbf{k}$ makes it difficult to perform assignment to the latent variables while also estimating the model parameters. The problem is solved by employing a variational EM procedure, which iteratively learn the model, alternating E and M-step. The E step aligns all bags of features to grid windows, to match the bags' histograms, inferring , i.e., were each bag maps on the grid. In the M-step we re-estimate the counting grid so that these same histogram matches are better. The procedure is illustrated with algorithm 1; $\pi_{\mathbf{k},z}^{old}$ is the counting grid at the previous iteration.

Even for large corpora the learning algorithm converges in 70-80 iterations, which sums up to minutes for summarizing corpora of

[1]Which may or may not be in the top three

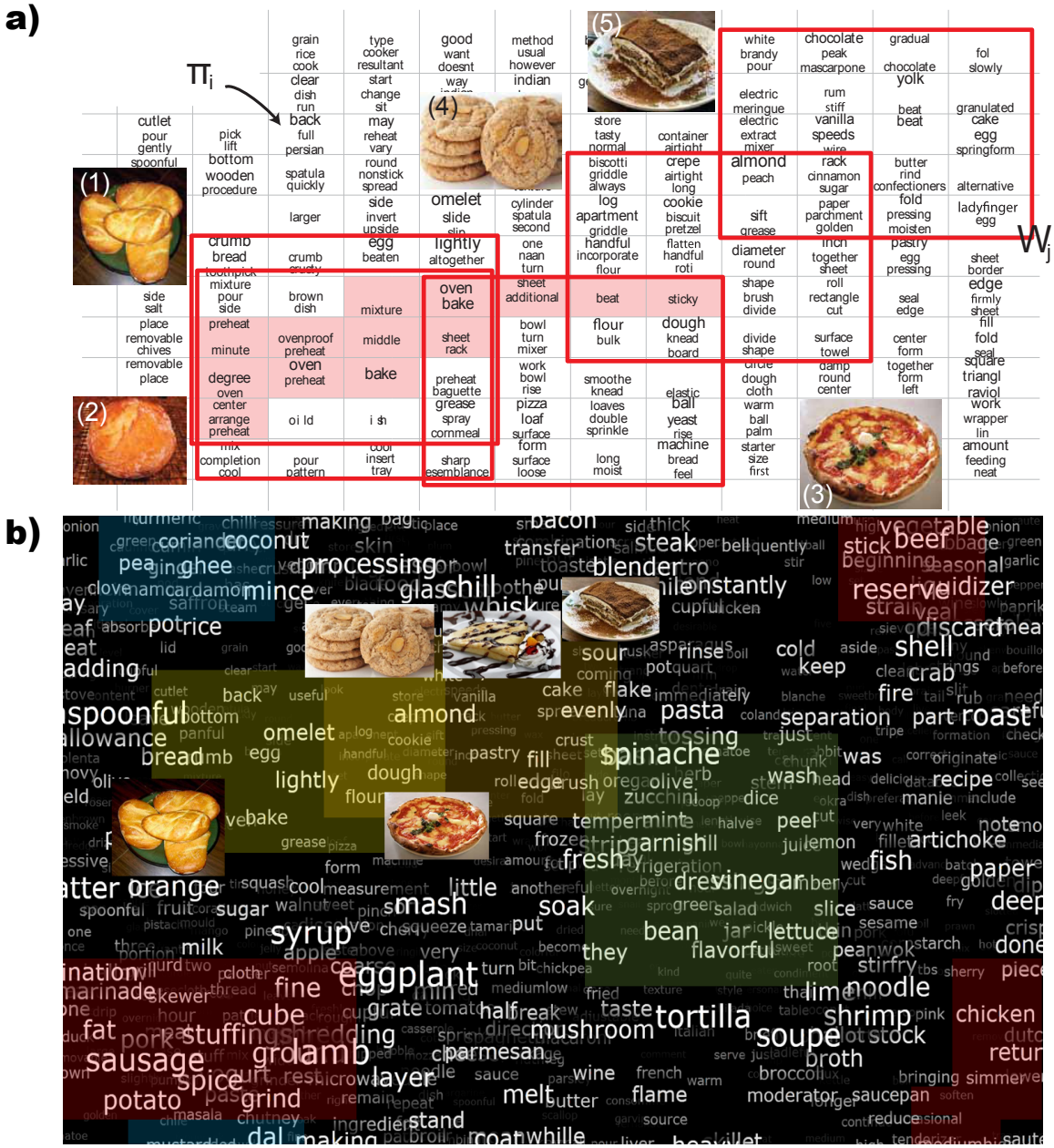


**Figure 1: a) A particular of an area of a counting grid $\pi_i$ learned over a corpus of recipes. In each cell we show the 0-3 most probable words greater than a threshold. The area in shaded red has $\pi('bake') > 0$. b) The interface built upon the (whole) counting grid shown in panel a) (here shaded in yellow). We also highlighted areas relative to spices (blue), vegetables (green), meats (red).**

over 40K documents.

As this EM algorithm is prone to local minima, the final grid will depend on the random initialization, and the neighborhood relationships for mapped documents may change from one run of the EM to the next. However, in our experience, the results always appear very similar, and most of the more salient similarity relationships are captured by all the runs.

More importantly, a majority of the neighborhood relationships make sense from a human perspective and thus the mapping gels the documents together into logical, slowly evolving (in space) themes. As discussed below, this helps guide our visual attention to the subject of interest.

## 3. COUNTING GRID AS A USER INTERFACE

In order to use the counting grid as an underlying representation in a powerful UI, we found three improvements necessary.

### 3.1 Optimization algorithms

Given the considerations in the introduction, the quality of em-

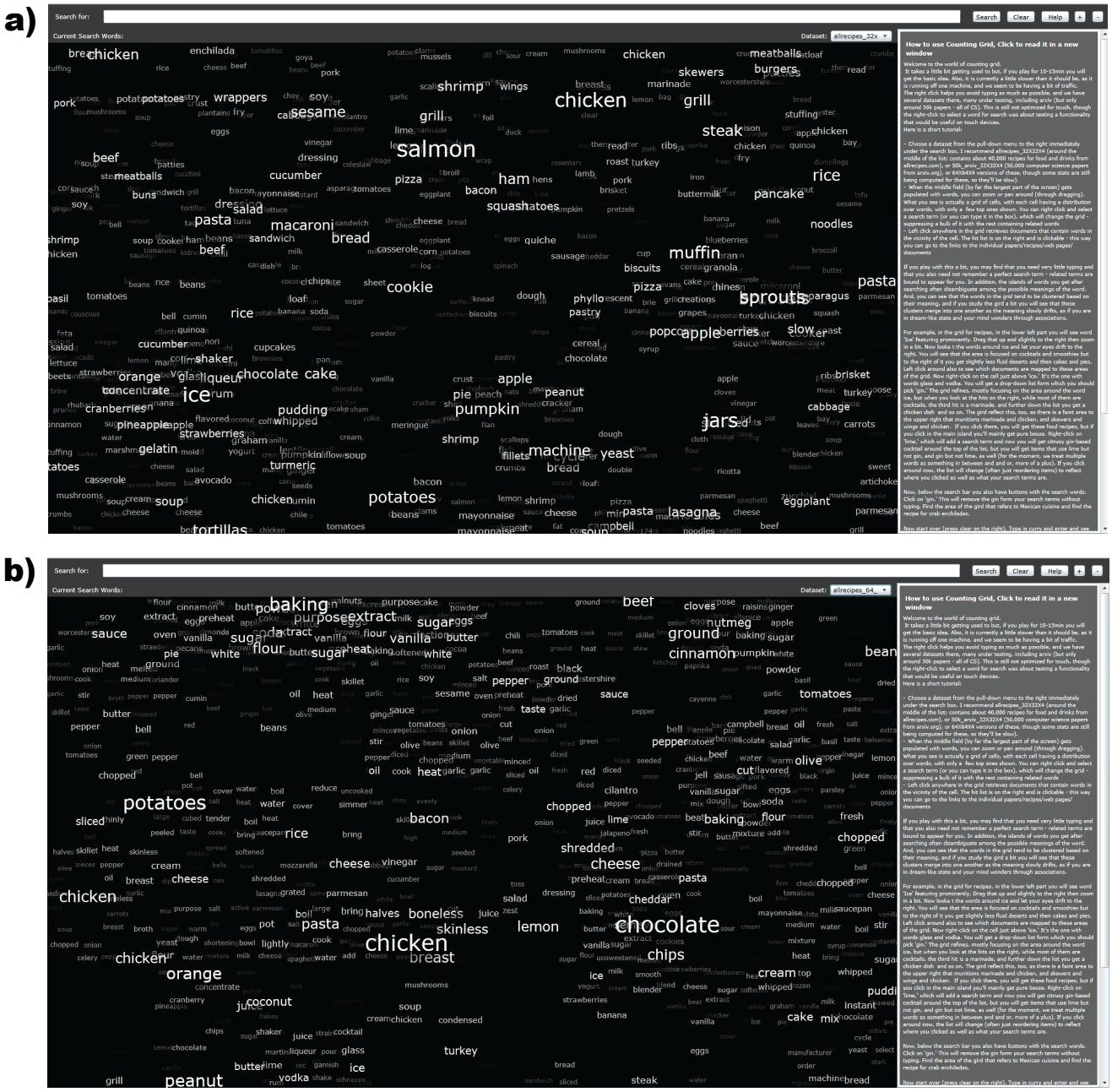


**Figure 2: Interface: a) Counting Grids b) Distance Embedding + Keywords.**

bedding can have a dramatic effect on the user experience. The CG model is more directly tied to the goal of visualizing higher order statistics in word usage patterns than previous models: It literally attempts to lay the words out so that nearby words can be found commonly in the documents (and even in the intersection of highly related documents). Thus the direct optimization of data likelihood should get us good embeddings. However, there are no globally optimal likelihood optimization methods for this model. Fortunately, the basic Em model derived in [6] does at least provably converge to a local minimum. Furthermore, for the purposes of the browser we tested here, we experimented with various ways of escaping local minima, such as sampling methods, random restarts, online learning/gradient descent, and found that the nicest grids with highest likelihood tend to be created by a multiresolution approaches.

In a first approach an $8 \times 8$ grid is first estimated using the $5 \times 5$ mapping window size. The grid is then upsampled by replacing each cell with a $2 \times 2$ set of cells with the same distribution. Then the EM learning of this $16 \times 16$ grid is continued using the same size of the mapping windows ($5 \times 5$) until convergence. This process is then iterated to the desired size of the grid.

In a second approach we kept fixed the grid size, progressively reducing the window size every 10 iterations until we reached the

desired window size.

We found that these multi-resolution approaches create longer thematic shifts and fewer boundaries among areas, which is generally more pleasing to the eye and makes it easier for the user to learn "the lay of the land." We believe that further improvements in optimization algorithms may create dramatically better results, esp. for large datasets.

## 3.2 Pan-zoom-click-search interface to a CG

The interface, shown in Fig. 2-4, allows several modes of interaction with the data and the grid. The grid itself is rendered so that the font size denotes the local weights of different words directly imported from the model. The weights essentially indicate how likely the words are in context of other nearby words. We have implemented a fast pan-zoom interface for exploration of the grid in Silverlight (Fig.3 shows the zoom). A click (or a tap on touch devices) shows the set of documents whose mapping windows overlap the point we clicked on. The list is shown on the right without changing the grid view. The grid can be filtered in two ways: by typing the search term in the search box, or by simply selecting a word (right click on long tap). Two search results are illustrated by Fig.4: in panel "a" memory, in panel "b", forest (see the text box on the top of the interface).

Assuming that a very specific search goal with a well formulated query cannot be aided much by dataset summarization and diversity exposure, we did not test the counting-grid representations primarily on such tasks. Instead, we have made our interface as close to traditional search-based interfaces as possible for such situations: The user can enter the search terms and the results will be presented in the list on the right hand side of the interface. However, through grid filtering described above, our interface also provides a diversity viewing experience that aims to expose the user to themes related to a specific successful query, as well as a summary/grouping of relevant content for less specific queries and summary, organization and visualization of the entire dataset for multi-objective or free-form browsing experience. Importantly, the counting grid representation combined with the pan-zoom-click-search interface enables a unified way of data consumption across these levels of granularity of user interest. For example, a high-quality query that results in high relevance of returned items will filter the very same grid representing the entire dataset, with the effects shown in-place, so that gradual removal of search query words will expand the scope till the entire dataset is shown. As the relative positioning of topics/themes stays fixed through this experience, moving back and forth among different search goals with possibly varying levels of specificity does not throw the user out of context, which in traditional interfaces poses a barrier for multifarious search and makes the user focus and organize their tasks linearly, rather than in parallel. Perhaps most importantly for the diverse application of the ideas presented here, the user interface is created automatically from the dataset as the input, using an unsupervised machine learning algorithm, and the result can in principle be refined by professional curator/designer or collaboratively by users, who can add their content or labels anywhere in the grid.

# 4. EVALUATION

We evaluated our interface in several ways. First, we were curious to see how much the direct optimization, in maximum likelihood sense, of embedding word sharing patterns aids the visualization of higher order co-occurrence statistics, and if these improvements indeed yield to increased speeds of resulting word cloud

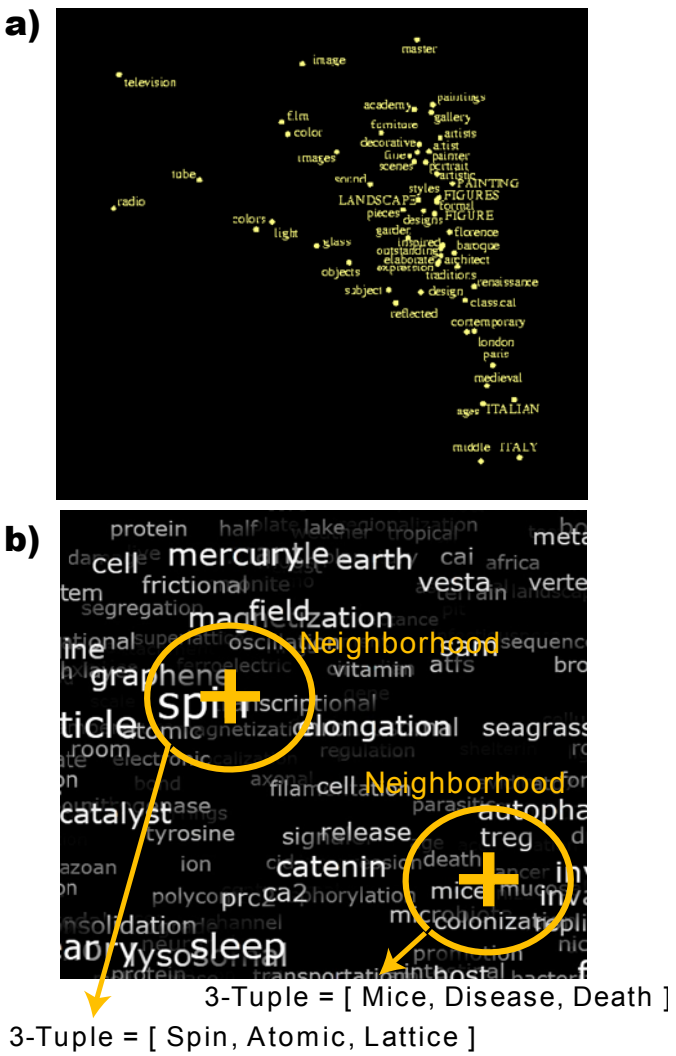


**Figure 5: a) A word embedding produced by an euclidean embedding method. b) The process of tuple sampling: A position is randomly picked on the grid and words are sampled from a neighborhood.**

skimming. As these results indicated a clear advantage of counting grids over the alternatives, we next investigated the amount of gleaned information during a short exposure to the data through our interface and compared this directly with the state-of-the art, but traditional web site interface to the same data, as such comparisons in the past tended to not show a quantifiable advantage of word clouds over simple search interfaces, while at the same time the user surveys usually showed that users like word clouds and are under the impression that the clouds may aid them in goal-free exploration of the content.
In all the experiments we employed the multiresolution approach of Sec. 3.1 to learn the grids, removing stop-words and applying the Porter-stemmer algorithm [9].

## 4.1 Word combinations at random focus areas: Numerical comparisons

One of the immediate goals of CG optimization is to create a

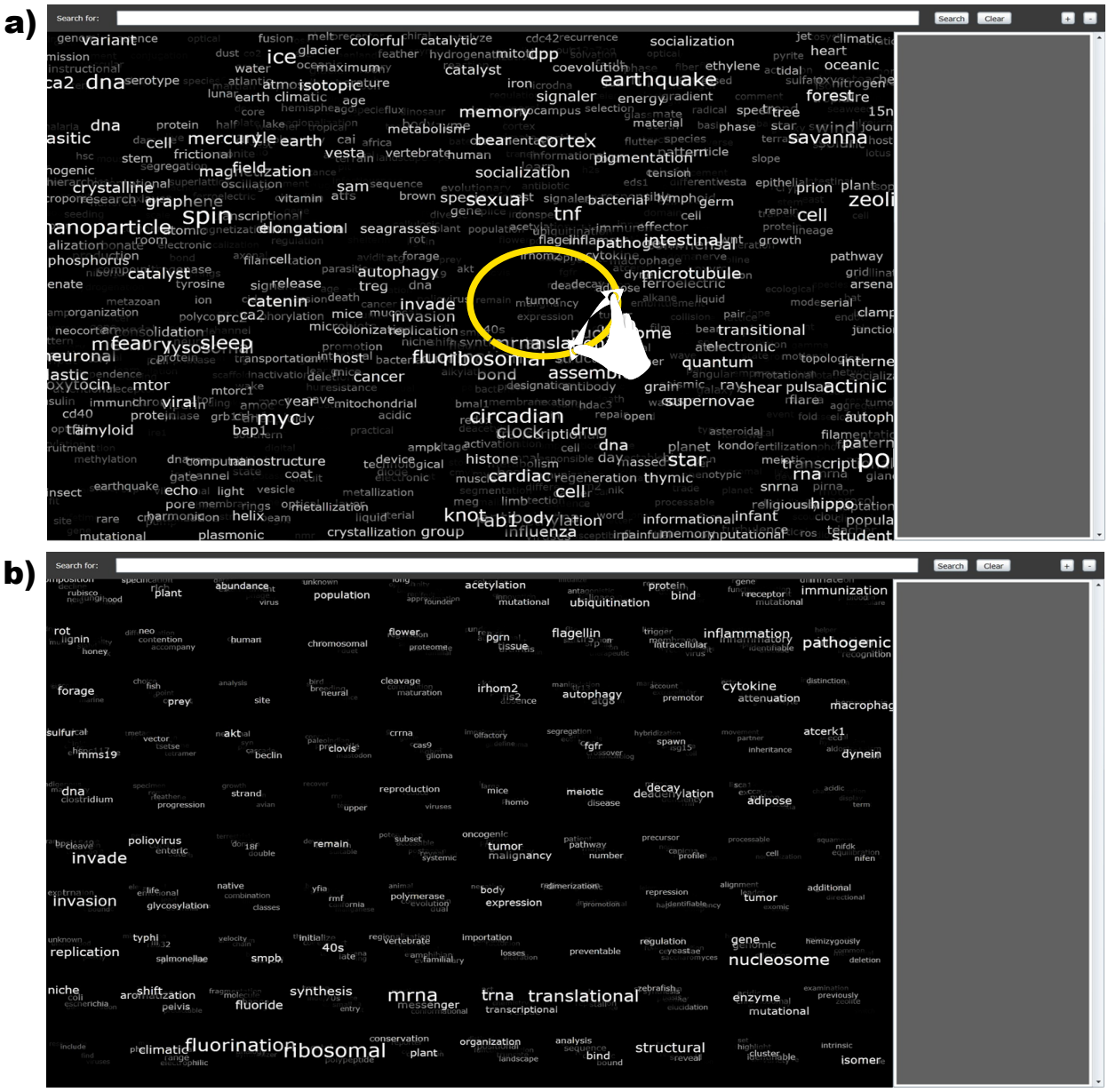


**Figure 3: Zoom: a) A counting grid learned using Science magazine papers and reports. The user can zoom until visualizing the top words of each source (panel b)**

visualization in which high order statistics of many word combinations can easily be visualized: In any local area of the grid, the words seen in the neighboring cells should "go together" so as to make the consumption of the grid easier. This aspect of the counting grids can be quantified directly without user studies, through hundreds of grid sampling steps.

In each step, a "neighborhood" in the window picked uniformly at random [2], and then $k$ words are drawn from that window according to the local word distribution. This sampling process is illustrated by Fig.5b.

Then these $k$-tuples are checked for *consistency* and *diversity* of indexed content. The consistency is quantified in terms of the average number of documents from the dataset that contained all $k$ words selected, while the diversity of indexed content is illustrated through the cumulative graph of acquired unique documents as more and more $k$-tuples are sampled and used to retrieve documents containing them.

We would expect that the CG model should show good consistency of words selected this way as the model is in fact optimized

[2] the curves look very similar for $3\times3$ to $7\times7$ window choices even though the grids were learned using assuming that each document maps to a $5 \times 5$ window

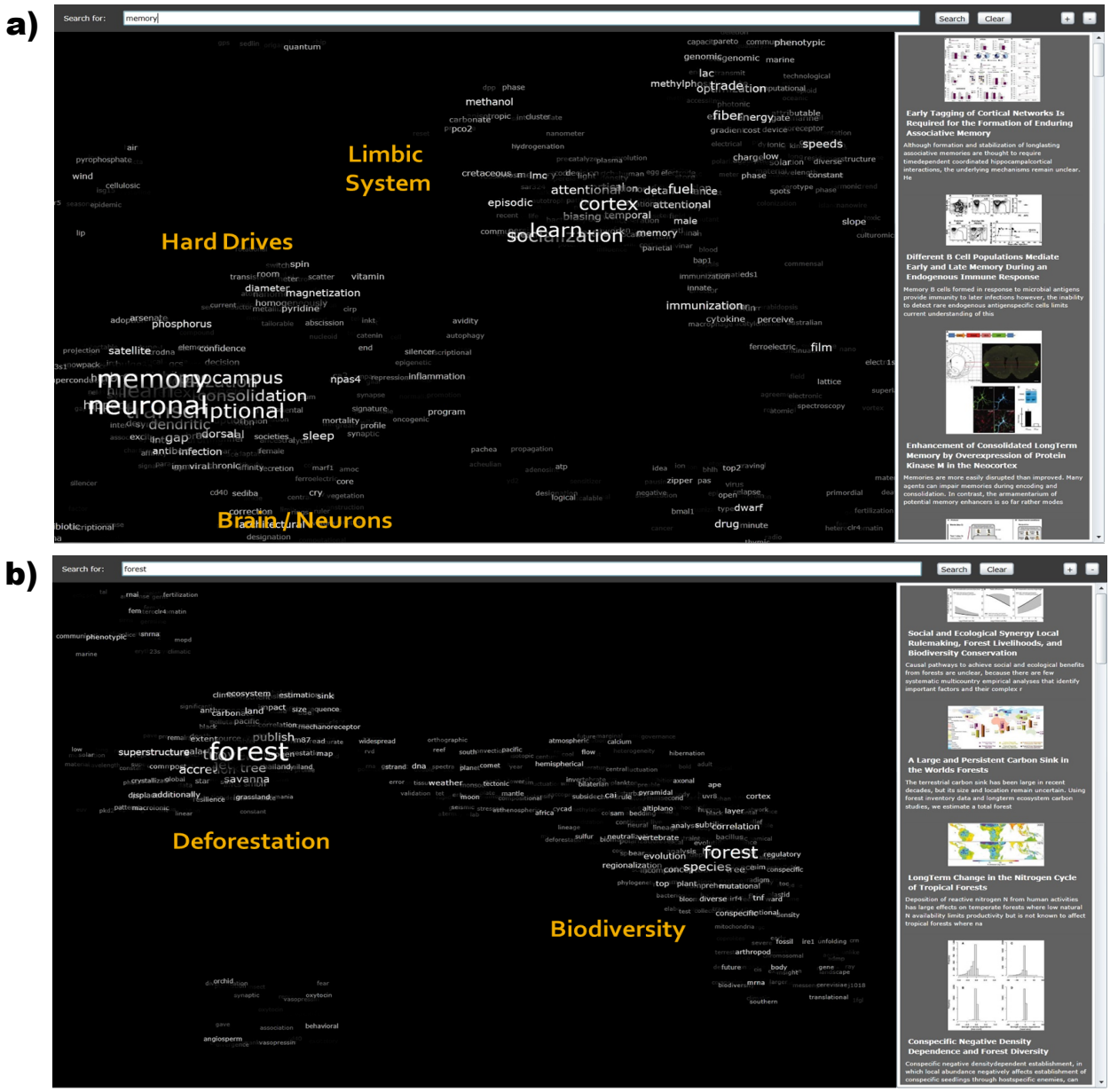


**Figure 4: Search results are presented as (non contiguous) islands on the grid, where different islands capture different semantics of keywords. For example, a) search result of the word "memory" reveals three islands related to computer memory, brain memory and the limbic system. Analogously b) search for the word "forest" revealed an island about deforestation and one about biodiversity in forests. By interacting with these islands the user can filter out unwanted results, or discover new things.**

so that documents' words map into overlapping windows, and so through the positioning and intersection of many related documents the words should end up being arranged in a fine-grained manner so as to reflect their higher-order co-occurrence statistics.

To the best of our knowledge there is currently no other technique that attempts to perform similar optimization, so we compare here with an approach based on previous techniques that achieved visually most similar arrangements, at least at a first glance (see Fig.2).

Some previous embedding techniques proposed word embedding based on pairwise distances, or joint embedding of words and the documents based on document-document and document-word distances [11, 4]. The problem with these approaches is that each word is assigned to a single location but certain highly informative words still assume multiple meanings in different contexts. For example, the word "memory" in the corpus of Science magazine papers can be found in articles on neuroscience, but also in immunology (immune memory of the adaptive immune system), device memory, as well as in quantum mechanics and occasional computer science papers. This would make such a word a nexus

| Corpus | # Docs | # Words | Tokens | Notes |
|---|---|---|---|---|
| Science Magazine | 36K | 24K | 2.0M | Papers and Reports |
| Allrecipes | 43K | 4K | 10M | |
| Arxiv | 25K | 31K | 2.3M | Computer Science |
| IMDB | 18K | 25K | 0.9M | Popular movies |

**Table 1: Statistics of the four corpora considered**

of several different clusters, making the browsing confusing in that area. Things are worse given that there are in fact many such words, and the attempts of embedding into 2D in this way usually collapse. Another promising approach is to simply focus on document embedding and then show representative words from nearby documents in the plane [3]. The problem here is that most embedding methods create a lot of empty space among clusters, which leads to dramatic under use of screen real estate (see Fig.5a).

Nevertheless, we embark on this approach to build a reasonable baseline for our method by further deforming the neighborhood relationships are maintained but the grid is denser (otherwise, this method would suffer on diversity measures described above). This baseline is further aided by making an effort to avoid local word repetitions which further reduce the information content of the grid and thus the diversity measure above. Fig. 2a shows the best so obtained embedding for allrecipes.com data, containing 43k recipes. Although at a first glance the two visualizations share a lot of common qualities, the sampling experiments show a dramatic difference in favor of CG on four different datasets, all approximately 50K in size: Science Magazine articles from the last 10 years, all of arxiv CS articles, allrecipes.com, and the most popular movies from IMDB. Details of each dataset are reported in table 1.

As shown in Fig.6 the more traditional distance embedding + keyword spraying approach matches, more or less, the quality of CG when we sample for word pairs (k=2). However, as this approach, or any other in the literature does not attempt to directly capture higher level statistics of word usage, even though the general clusters look meaningful at a first glance that capture grow structure of the data, the fine grained local structure of CGs much better captures higher order correlation, with this advantage typically growing with $k$. One outlier seems to be the most diverse Science dataset with the richest vocabulary. The curves in Fig. 6 are pretty close, but Fig. 7a which shows the diversity of the indexed information explains the difference. Fig. 7b, shows the gradient of the last curve of Fig. 7a.
An embedding of words that creates the same trivial combinations of words in many areas of the grid (e.g. {salt, paper, sprinkle} would boost the fraction of dataset covered by this triple. However, the number of new documents would then not grow. In case of counting grid, not only are the k-tuples meaningful, but they are diverse and with repeated jumping over the grid more and more content is being retrieved, which is the combination we want in a user interface meant for summarizing, browsing and retrieval.

## 4.2 High speed multifarious search and the extent of collateral information gleaned

As we discussed in the introduction, the main motivation in research on visualizing datasets by mapping documents and/or displaying word clouds is in the potential ease in understanding the extent of the dataset, locating topics of interest quickly when these interests are not well defined, as well as accidental discovery of interesting and useful information [1] that is somewhat related to the original goals of the information seeking process.

Here we test the ability of users to rapidly gain insight both into specific and broad topics which are either directly or indirectly related to a mix of topics of interest, as well the collateral information gleaned in the process.
The traditional search paradigm would force us to try to look for this research linearly, focusing on one are at a time, getting new ideas for search only once we read the discovered papers. The counting grid visualizations with orders of magnitude more words than usual tag clouds and at a same time much denser and better organized embedding of relevant documents may (and did) enable us do some of these investigation rapidly and in parallel, jumping from topic to topic as the links are revealed. We assume, of course, that such multifarious search, where a variety of topics are of interest, some at a high level, and others needing to be explored in depth is often attempted by Internet users in a variety of tasks, and we focus here again on the allrecipes dataset.

We created a questionnaire with 60 questions of various specificity about the contents of the dataset by repeatedly sampling recipes form the dataset and formulating questions at different level of description depth, like "Are there Indian dishes here?", "Are there crepe recipes in this dataset?", "Are there savory recipes?", "Do any recipes use zucchini?" etc. Then we added several control questions for which we knew that they referred to items not covered by the dataset, like "Wine reviews" or "Cheese platters" or "Cooking book reviews". We expect that the users' performance on this task should be predictive of the experience they would have with our tool in many real world scenarios.
We compared the CG interface with the allrecipes web's own professionally and community-curated easy-to-use and powerful interface, which includes a modern search engine, various categorizations of the data, user-supplied votes and labels, etc.

We recruited seven subjects for the study and told them that they would be asked to answer a series of questions, including a list of ten that we read to them, and that they had 3 minutes to find out as many answers as they can using one combination of a dataset and an interface at a time.

We told the users that they would do this for two such combinations and we convinced them that the two combinations differ both in the extent of the data and the user interface, and that other than the initial 10 questions, the questionnaires would also be randomly related. However, to avoid issues with comparisons of different questionnaires and datasets on a small sample of users, we in fact varied only the interface, and used the same dataset asked the same questions, but placed the competing interface second in the study, where it would presumably have an advantage over our method if the users would never the less be inclined to look for answers to the questions beyond the initial ten questions. We hoped that the limited amount of time provided for the task would minimize that advantage anyhow, as our preliminary tests on the authors and pre-test subjects indicated that more than five minutes were need to perform all the searching necessary to cover a large fraction of questions if the user is searching based on their memory of the entire questionnaire. In addition, we found that no subject was able to find information relevant to more than about half of the questions asked, indicating again that there was not enough time for the traditional interface to gain significant unfair advantage over our interface. No single question was answered correctly by all users, except for the control questions, for which the real answer was no (There were no wine reviews in the data, etc.), indicating that they

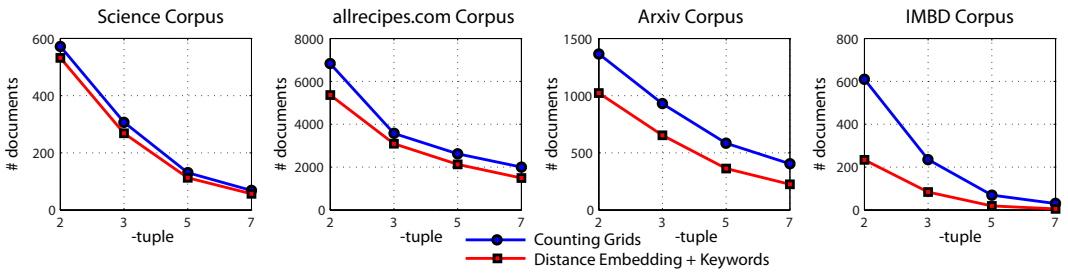


**Figure 6: Consistency results**

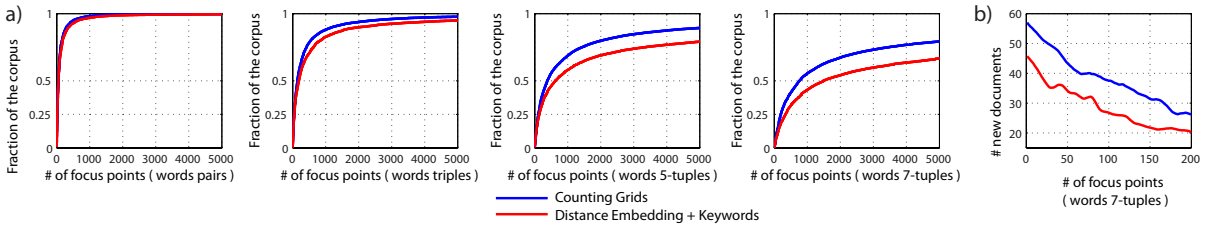


**Figure 7: Diversity results**

were giving us honest representations of what they remembered. All seven users performed better using counting grids ($p<0.01$), with the average gain in the number of questions answered of 60% over the allrecipes.com interface, despite the potential advantages that the latter may have had due to order of testing. The ability to glean collateral information beyond the 10 questions to which the users were primed was certainly biased by users' food preferences or familiarity with cooking styles. Only our one Chinese test subject detected traces of the Chinese cuisine in the counting grid based on the combination of ingredients much more typical of the Chinese cuisine; a quick click there indeed revealed Chinese dishes he had in mind. The types of meat and vegetables the users found or did not find in the dataset typically correlated with their preference for these foods.

However, for all users in this small study, the intersection of their preferences with the questions asked was enough to provide enough answers in order to see the difference between the two interfaces. Interestingly, the percentage of answered questions varied more widely after using the allrecipes.com standard interface (as low as 22% and as high as 46%) than for CG interface (42% - 51%), which provides another indication of the interaction between the users' own memory and the CG. Using the standard search interface the users could not remember or think to explore further items of low interest to them, even after seeing recipes that could provoke further investigation. But the word associations in the CG interface seemed to more readily enter their visual field and remind them of the task defined by the initial questions. Results are summarized in Fig. 8.

In post-test interviews, all users indicated preference for the CG interface for the task of rapidly discovering lots of information as well as for organizing the data. They could simply "see much more in parallel" in the CG interface, and could often recognize recipes just based on the words in the grid and without opening any of the documents mapped in the area. They also indicated that they had a better understanding what data was exposed by the CG interface, while the boundaries of allrecipes.com interface seemed uncharted and the data thus appeared potentially vast (even though the number of recipes was approximately the same). When asked for a subjective estimate of how much information they encountered while using the CG compared to the standard interface, they reported factors of 2-5, which are either inflated subjective estimates (compared to the measured factors which varied between 1.2 and 1.9), or they indicate that the users saw much more of the content related to their food interests in addition to the content to which we primed them to look for. The latter possibility would be in line with previously observed difficulties in measuring the diversity of information the user accesses during data exploration.

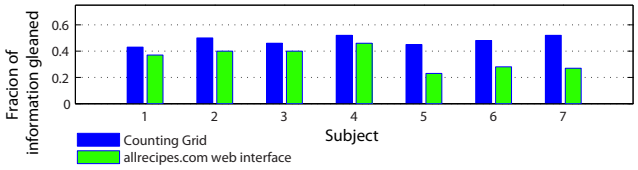


**Figure 8: Information gleaning experiment**

## 5. CONCLUSION

To the best of our knowledge, the counting grid visualization we presented here is the first system that directly optimizes for simultaneous presentation of word co-occurrence statistics of various orders well beyond the usual pairwise embedding. This is accomplished through a dense word and document embedding that facilitates a visual browsing and search paradigm that can more naturally rely on the cognitive processes we employ when we scan visual scenes as well as the ones that guide visual recognition of words in skim reading. We have shown that the CG representation tends to display words that go together in almost any location in the visual field, and that by sampling different local combinations of words we tend to identify a larger fraction of the dataset and in a more diverse manner across locations than we can achieve using standard embedding methods to display large number of words from embedded documents. We also find that this increased semantic order does indeed facilitate faster visual processing of the word map, as well as faster memorization of the word distributions in word

spotting experiments. In addition to data organization, the CG visualization also facilitates interesting patterns of partial document consumption. By spotting several related words, the user is reminded of the knowledge they already have, and may not even need to open relevant documents. As described in the grocery shopping case study, in such cases the effect is akin to parallel skim-reading of hundreds of documents that contain the word combination to narrow down on a known common theme and extrapolate (remember) the rest of the document to the extent needed by the user. In addition, surprising combination of words in the area the user is interested in can lead to serendipitous discovery of new documents to be studied in detail.

From the perspective of word/tag cloud usability research perhaps the most exciting result comes from our preliminary experiments on multifarious search and serendipitous data exposure that show that thousands rather than dozens of words on the screen can still be consumed by the user and that the extent of the data explored this way is high enough that the differences can be quantified in user studies.

However, despite encouraging preliminary results, a lot about counting grid representations and interface design remains to be studied. We found that the quality of the embedding of high order statistics matters, yet we know from our experiments that the current algorithms are prone to local minima. Thus it remains to be seen if the document packing can be done more optimally in the maximum likelihood sense and if such improved grids would provide even better local word combinations that would be even easier to browse/search. We have experimented with a wide variety of medium-sized datasets containing tens of thousands of documents. It remains to be seen what the best way would be to scale this experience to very large datasets. Interface refinements can play a big role, too. For example, in our three-tiered approach to visual searching over the grid – visual scanning, filtering by word seen in the grid, or filtering by a typed word not yet spotted – the last modality tended to be avoided by users to unreasonable levels because it was perceived to be at odds with the smoother experience of combining visual scanning with mouse/touch actions.

# 6. REFERENCES


[1] P. André, M. C. Schraefel, J. Teevan, and S. T. Dumais. Discovery is never by chance: Designing for (un)serendipity. In *In C and C Š09*. ACM, 2009.

[2] C. A. Becker. Semantic context effects in visual word recognition: An analysis of semantic strategies. *Memory and Cognition*, 8(6):493–512, 1980.

[3] B. Fortuna, M. Grobelnik, and D. Mladenić. Visualization of text document corpus. *Special Issue: Hot Topics in European Agent Research I Guest Editors: Andrea Omicini*, 29:497–502, 2005.

[4] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. In *Advances in Neural Information Processing Systems 17*, pages 497–504. MIT Press, 2005.

[5] H. Intraub. The representation of visual scenes. *Trends in Cognitive Sciences*, 1(6):217–222, Sept. 1997.

[6] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. In *UAI*, pages 547–556, 2011.

[7] M. A. Just and P. A. Carpenter. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87(4):329, 1980.

[8] A. Paivio. *Mental Representations: A Dual Coding Approach (Oxford Psychology Series, 9)*. Oxford University Press, 1990.

[9] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[10] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, pages 372–422, 1998.

[11] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[12] J. Schrammel, M. Leitner, and M. Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of CHI '09*, 2009.

[13] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings on Information Visualization*, pages 51–58, 1995.