Learning recipe ingredient space using generative probabilistic models

Vladimir Nedović
Science Hub
Amsterdam, The Netherlands
vladimir.nedovic@science-hub.com

Abstract

In this paper, we demonstrate preliminary experiments using generative probabilistic models on recipe data. Recipes are reduced to lists of ingredients and analyzed in a bag-of-words fashion. We first visualize the highly-dimensional ingredient space and map it to different world cuisines. Latent Dirichlet Allocation (LDA) and Deep Belief Networks (DBN) are then used to learn generative models of ingredient distributions and produce some novel ingredient combinations. First results demonstrate the feasibility of the approach and point to its promise in recipe improvization.

1 Introduction

To the first approximation, a recipe consists of an ingredient list and the accompanying cooking instructions. In [Buykx and Petrie, 2011], the authors show that splitting recipe content into distinct blocks is rated best by the cooks who use the recipe. In addition, ingredient amounts are shown to be more useful within the method instructions than when presented together with the ingredient overview. Therefore, ingredient list and instruction sections can safely be addressed individually. In this work, we analyze the ingredient space of different recipes, resorting only to their ingredient lists.

A lot of research in the past several years focused on the recipe method, analyzing its text or augmenting its content using other modalities. We take a different approach. We observe that digital recipe texts, even in the ingredient overview part, are still rather static and isolated from each other regardless of similarities between the dishes they represent. Therefore, we aim to analyze ingredient lists at word level to provide an entry-point in recipe discovery.

Treating ingredients individually can lead to establishing ingredient correlations and recipe similarities. For instance, the former could allow for ingredient substitutions, whereas the latter could enable combining individual recipes for a single dish. In addition, dishes could be visualized in context, in terms of flavors combined, geographical region, and so on. For example, Wikipedia lists 28 different kinds of meatball dishes, even excluding the regional variations within countries. Given that all these dishes share the same ingredient base, they could easily be connected, and together give rise to a potentially new dish variant. In this paper, our goal is to demonstrate preliminary experiments which lead in that direction.

2 Related work

There has been an increasing body of research lately concerning automation related to recipe information and cooking activities. Beside many groups and companies taking part in a *Computer Cooking Contest*[1], workshops are being organized in conjuction with AI and multimedia conferences. In addition, we are seeing recommendation systems based on ingredient networks [Teng *et al.*, 2011] as well as attempts to quantitatively validate the food pairing hypothesis [Ahn *et al.*, 2011]. All these activities indicate an increasing interest in using computers in the kitchen, as well as increasing awareness of the importance of cooking in everyday lives.

Within the body of research, some approaches provide multimedia enhancements of textual recipe information. This is aimed at facilitating the process of cooking or at fixing some undesirable behavior [Wiegand *et al.*, 2012]. Whereas some methods focus on supporting a person while cooking [Hamada *et al.*, 2005; Ide *et al.*, 2010], others help people who follow a specific diet [Brown *et al.*, 2006; Chi *et al.*, 2007]. In contrast to these approaches, which augment existing textual information, we take a step back and concentrate on finding the limits of analysis of recipe texts, more specifically their ingredient lists.

The second group of recent approaches build on standard knowledge discovery (KDD) systems, adapted to recipe data. For example, the authors in [Gaillard *et al.*, 2012; Dufour-Lussier *et al.*, 2012; Mota and Agudo, 2012] focus on case-based reasoning (CBR) methods. CBR is the method of solving problems based on previous solutions to similar problems. However, this process mostly involves data ontologies and expert knowledge. We follow a different path. Instead of imposing structure on data, we aim to discover any structure that may be present in recipes using machine learning approaches.

We mostly draw inspiration from [Ahn *et al.*, 2011] and [Teng *et al.*, 2011]. However, where [Ahn *et al.*, 2011] analyzes foods at the level of flavor compounds, we limit ourselves to the level of individual ingredients. In contrast to

[1] `http://computercookingcontest.net/`

[Teng *et al.*, 2011], which computes complement and substitution networks based on co-occurences, we wish to go a step further and find longer-range ingredient and dish relationships as well. We aim to do so by employing appropriate pattern recognition methods.

In the machine learning literature, many document analysis methods rely on extensions of Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003]. LDA is a generative probabilistic model that represents each document as a mixture of a small number of hidden topics. In addition, each document word can be assigned to one of the document's topics. These methods can be easily applied to the recipe domain, as has been done in [Mori *et al.*, 2012], where the text of the recipe method is being analyzed. We also use LDA, but start by applying it to ingredient lists only, aiming to discover latent ingredient bases underlying specific recipes.

3 Generative Probabilistic Models

In this section, we give an overview of the two generative, latent variable models that we use for modeling ingredient distributions. Those are Latent Dirichlet Allocation (LDA) and Deep Belief Networks (DBN).

3.1 Latent Dirichlet Allocation

LDA is a generalization of the earlier Probabilistic Latent Semantic Analysis (PLSA) [Hofmann, 2001]. Both are well-known latent variable models for high dimensional count data, especially text in a bag-of-words representation. In this representation, D documents are each represented as a vector of counts with W components, where W is the number of words in the vocabulary. Each document j in the corpus is modeled as a mixture over K topics, and each topic k is a distribution over the vocabulary of W words. Each topic, ϕ_k, is drawn from a Dirichlet distribution with parameter η, while each document's mixture, θ_j, is sampled from a Dirichlet with parameter α. For each token i in the corpus, a topic assignment z_i is sampled from θ_{d_i}, and the specific word x_i is drawn from ϕ_{z_i}. The generative process is thus:

$$\theta_{k,j} \sim D[\alpha] \qquad \phi_{w,k} \sim D[\eta] \qquad z_i \sim \theta_{k,d_i} \qquad x_i \sim \phi_{w,z_i}.$$

Exact inference (i.e. computing the posterior probability over the hidden variables) for this model is intractable [Blei *et al.*, 2003] and thus a variety of approximate algorihtms have been developed. Ignoring α and η and treating θ_{kj} and ϕ_{wk} as parameters, we obtain the PLSA model, and maximum likelihood (ML) estimation over θ_{kj} and ϕ_{wk} directly corresponds to PLSA's Expectation-Maximization (EM) algorithm.

Starting from the form of log-likelihood,

$$l = \sum_i \log \sum_{z_i} P(x_i|z_i,\phi)P(z_i|d_i,\theta)$$

we obtain the parameter updates via standard EM derivation:

$$P(z_i|x_i,d_i) \propto P(x_i|z_i,\phi)P(z_i|d_i,\theta) \tag{1}$$

$$\phi_{w,k} \propto \sum_i \mathbb{I}[x_i = w, z_i = k]P(z_i|x_i,d_i) \tag{2}$$

$$\theta_{k,j} \propto \sum_i \mathbb{I}[z_i = k, d_i = j]P(z_i|x_i,d_i) \tag{3}$$

These updates can be rewritten by defining $\gamma_{wjk} = P(z = k|x = w, d = j)$, N_{wj} the number of observations for word type w in document j, $N_{wk} = \sum_j N_{wj}\gamma_{wjk}$, $N_{kj} = \sum_w N_{wj}\gamma_{wjk}$, $N_k = \sum_w N_{wk}$ and $N_j = \sum_k N_{kj}$. Then,

$$\phi_{w,k} \leftarrow N_{wk}/N_k \qquad \theta_{k,j} \leftarrow N_{kj}/N_j.$$

Plugging these expressions back into the one for the posterior in Equation 1, we arrive at the update,

$$\gamma_{wjk} \propto \frac{N_{wk}N_{kj}}{N_k} \tag{4}$$

where the constant N_j is absorbed into the normalization.

3.2 Deep Learning

Since no exact inference is possible in LDA and PLSA, they have to resort to slow or inaccurate approximations to compute the posterior distribution over topics. This makes it difficult to fit the models to data. In addition, there are limitations on the types of underlying structure that can be represented efficiently by a single layer of hidden variables.

To that end, Deep Belief Networks have been introduced in [Hinton *et al.*, 2006]. DBNs are generative probabilistic networks, composed of multiple layers of latent variables, which typically have binary values. They are usually represented by Restricted Boltzmann Machines (RBM) [Ackley *et al.*, 1985] at each layer. The layers of visible and hidden units are connected by a matrix of symmetrically weighted connections, optimized dring the learning phase.

Given an observed word count vector $\mathbf{v}$ and hidden topic features $\mathbf{h}$, let $\mathbf{v} \in \{1, ..., W\}^U$, where W is the dictionary size and U is the document size, and let $\mathbf{h} \in \{0, 1\}^F$ be binary stochastic hidden topic features. Let $\mathbf{V}$ be a $W \times U$ observed binary matrix with $v_u^w = 1$ if visible unit u takes on w^{th} value. The energy of the state $\{\mathbf{V}, \mathbf{h}\}$ is defined as follows:

$$E(\mathbf{V},\mathbf{h}) = -\sum_{u=1}^{U}\sum_{f=1}^{F}\sum_{w=1}^{W} M_{uf}^w h_f v_u^w - \sum_{u=1}^{U}\sum_{w=1}^{W} v_u^w b_u^w - \sum_{f=1}^{F} h_f a_f \tag{5}$$

where $\{M, a, b\}$ are the model parameters: M_{uf}^w is a symmetric interaction term between visible unit u that takes on values w, and hidden feature f; b_u^w is the bias of unit u that takes on value w, and a_f is the bias of hidden feature f. The probability that the model assigns to a visible binary matrix $\mathbf{V}$ is:

$$P(\mathbf{V}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp(-E(\mathbf{V},\mathbf{h})), \tag{6}$$

where Z is the partition function or normalizing constant:

$$Z = \sum_{\mathbf{V}}\sum_{\mathbf{h}} \exp(-E(\mathbf{V},\mathbf{h})) \tag{7}$$

The conditional distributions are given by softmax and logistic functions:

$$p(v_u^w = 1|\mathbf{h}) = \frac{\exp(b_u^w + \sum_{f=1}^{F} h_f M_{uf}^w)}{\sum_{q=1}^{W}\exp(b_u^q + \sum_{f=1}^{F} h_f M_{uf}^q)} \tag{8}$$

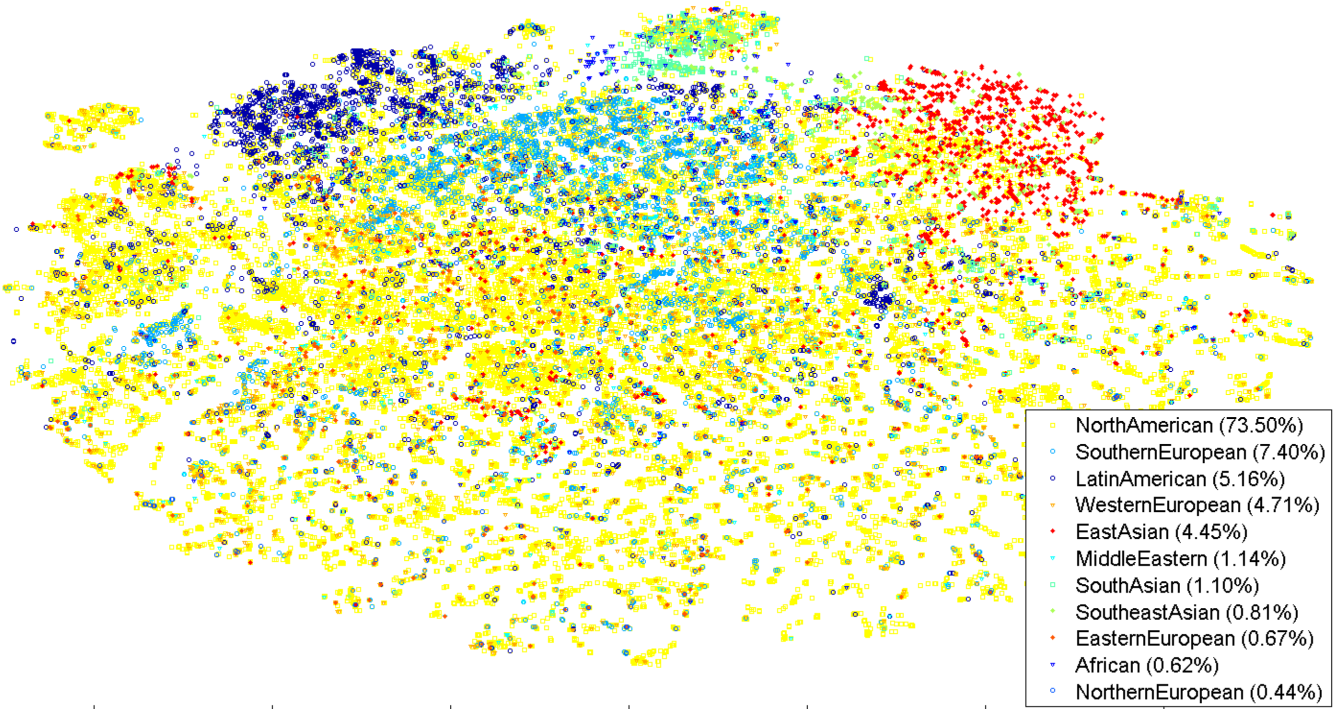


Figure 1: Visualization of ingredient space - a mapping to 2 dimensions using t-SNE. Different cuisines are represented by different markers. For each cuisine, the number in parentheses indicates the percentage of recipes that it covers in the dataset.

$$p(h_f = 1|\mathbf{V}) = \sigma\left(a_f + \sum_{u=1}^{U}\sum_{w=1}^{W} v_u^w M_{uf}^w\right). \qquad (9)$$

In document retrieval, a separate RBM is created for each document, and there are as many softmax units as there are words in the document. All the softmax units can share the same set of weights, connecting them to binary hidden units. Given a collection of N documents $\{\mathbf{V}_n\}_{n=1}^{N}$, the derivative of the log-likelihood w.r.t. parameters M takes the form:

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log P(\mathbf{V}_n)}{\partial M_f^w} = \mathrm{E}_{P_{data}}[v^w h_f] - \mathrm{E}_{P_{model}}[v^w h_f], \qquad (10)$$

where $\mathrm{E}_{P_{data}}[\cdot]$ denotes an expectation w.r.t. the data distribution $P_{data}(\mathbf{h}, \mathbf{V}) = p(\mathbf{h}|\mathbf{V})P_{data}(\mathbf{V})$, with $P_{data}(\mathbf{V}) = \frac{1}{N}\sum_n \delta(\mathbf{V} - \mathbf{V}_n)$ representing the empirical distribution, and $\mathrm{E}_{P_{model}}[\cdot]$ is an expectation w.r.t. the distribution defined by the model. Exact maximum likelihood learning in this model is intractable, and learning is usually done by following an approximation to the gradient of a different objective function, a process called contrastive divergence.

4 Learning recipe ingredient space

4.1 Dataset

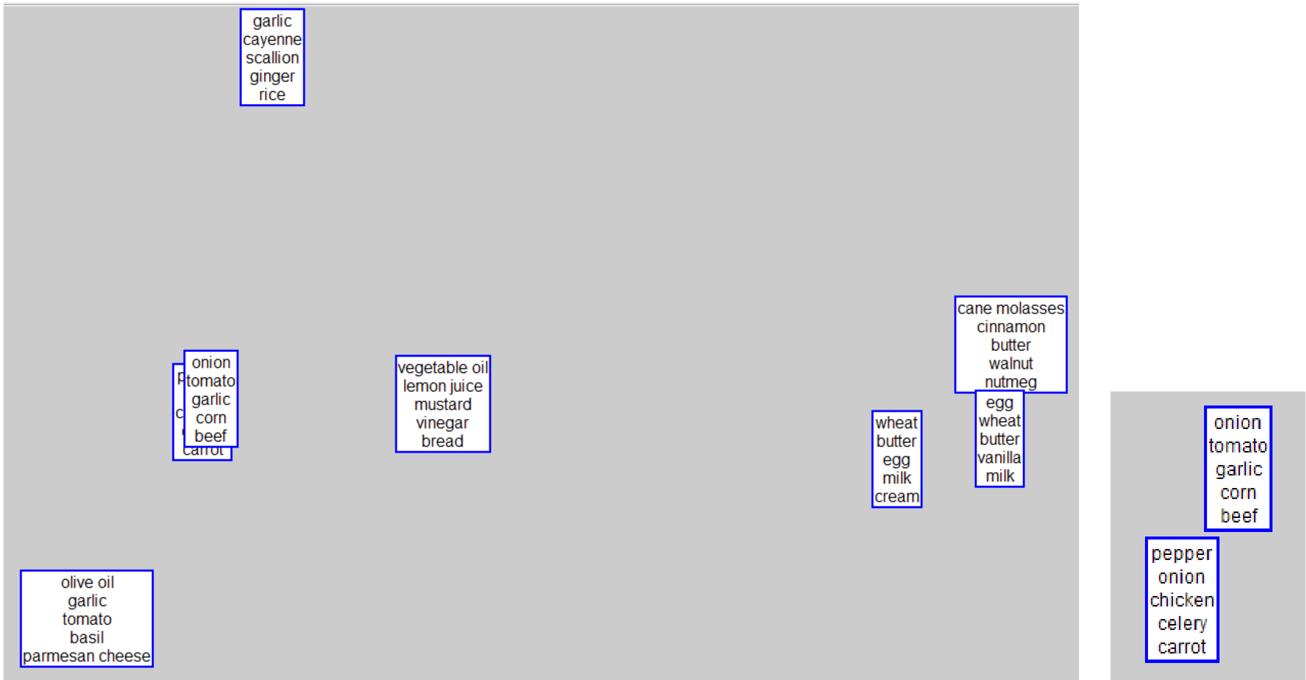
In this work, we aim to discover relationships, whether explicit or implicit, that may exist between different recipe vectors. For the preliminary experiments, we use the recipe collection of [Ahn *et al.*, 2011], which comes with more than 56000 recipes and 381 unique ingredients. The data was acquired by crawling three large recipe depositories, two American (*allrecipes.com*, *epicuirous.com*) and one Korean (*menupan.com*) (for parsing details, please see [Ahn *et al.*, 2011]). The recipes in the dataset are represented as ingredient lists only; therefore, we only consider the presence or absence of individual ingredients at this stage.

4.2 Cuisine mapping

To visualize the data and obtain some insight into its structure, we use the whole recipe corpus together with cuisine labels that are supplied with it. We utilize the technique of t-Distributed Stochastic Neighbor Embedding [van der Maaten and Hinton, 2008], which has shown promising results in visualizing the structure of high-dimensional data. Figure 1 shows a mapping of the ingredient space to different cuisines in two dimensions.

As can be seen from the figure, different cuisines are not equally represented in the dataset. In fact, North American recipes account for almost 3/4 of all the data. Nevertheless, certain conclusions can still be drawn even from this biased collection:

- South European, Latin American and East Asian recipes constitute very distinct dish groups;
- the above groups are nevertheless close to each other in the ingredient space;
- Asian cuisines (East, South and South East ones) are all connected into a separate cluster;



(a)

(b)

Figure 2: Latent Dirichlet Allocation. (a) Spatial visualization of 6 LDA topics. (b) Zoomed-in view of the two overlapping topics on the left, showing two different bases containing meat.

- Western European and North American groups cover the largest variety in ingredient space, overlapping with all other cuisines (although this effect is probably attributable to the bias in recipe distribution);
- there is a small group of Latin American dishes that is effectively closer to East Asian and South European cuisines than the Latin American one; same is true for a small group of South European dishes farther away from its base and closer to the North American tradition;
- Middle Eastern dishes sit between South European ones and East Asian ones;
- Eastern European dishes overlap with South European and North American ones; etc.

Therefore, different cuisines, which are essentially human constructs, indeed form distinct groupings in the ingredient space. This is observable, in only two dimensions, even when recipes are reduced to ingredient lists and only the ingredient presence/absence is considered.

4.3 Factor analysis

Latent variable models effectively project the data to a lower-dimensional manifold. However, the number of latent variables in such cases is not immediately apparent. Although many techniques exist for learning the intrinsic data dimensionality, most of them are very unstable, and we resort to an experiment with Factor analysis. The performance over different number of factors gives an indication of the number of latent variables for use in other models, such as LDA. For this purpose, we use the *Matlab Toolbox for Dimensionality Reduction* by Laurens van der Maaten[2].

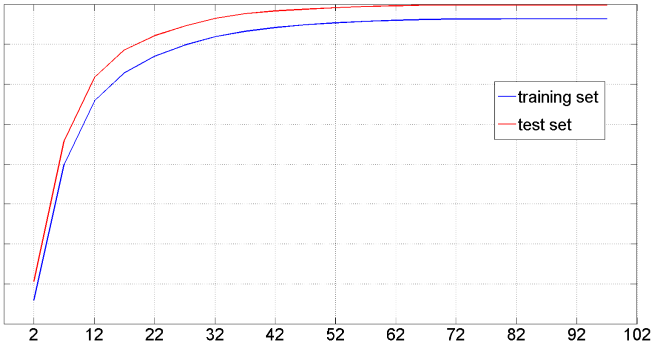


Figure 3: Factor analysis: log-likelihood vs. number of factors/dimensions used.

We split the data randomly into two halves, one used as the training set and the other as the evaluation set. For each different dimensionality, we learn the factor analysis mapping from the training data and compute the corresponding log-likelihood. That same mapping is then applied to the evaluation set, giving another log-likelihood value. This procedure is repeated for the number of factors ranging from 2 to 100, with a step of 5. The results are given in Figure 3, showing the log-likelihood leveling up after approximately 60 factors.

[2] http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

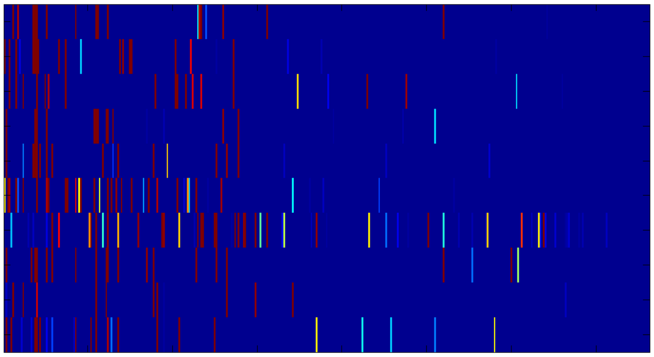


Figure 4: Ten new recipes sampled from the learned distribution given by a 3-level DBN.

4.4 Latent ingredient bases

Our goal is to analyze existing ingredient space and generate novel ingredient combinatons based on the learned models. We use LDA to project our highly-dimensional ingredient space to a smaller number of topics (i.e. *ingredient bases*) and then observe whether these bases make sense and how they are distributed. Since 60 or so dimensions, as given by factor analysis, would result in a cluttered figure, for visualization purposes we project to 6 topics only. The hyperparameters on the Dirichlet priors are set to $\alpha = 50/K$ and $\eta = 200/W$,: changing their values does not influence the results. We use the LDA package of [Griffiths and Steyvers, 2004] and show the results in Figure 2.

The figure shows the main ingredient bases maximally spaced apart on the horizontal axis: savory ones to the left and sweet ones to the right of the map. In addition to two "meat bases" on the left side, there is a base containing typical South-East Asian ingredients on top, and another one with common South European ingredients at the bottom of the figure. The ingredient base with common additives that affect acidity or fluidity of a dish is placed in the center of the map. The visualization implicitly shows the measure of how frequently certain (sets of) ingredients are used together.

4.5 Generating novel ingredient combinations

If we learn a generative probabilistic model from the ingredient data, we can also randomly sample it and observe the resulting ingredient combinations. These combinations will not necessarily correspond to those observed in the recipe corpus, but may represent completely novel varieties. In fact, one can imagine different parameter settings resulting in varying 'levels of combination novelty'.

For this purpose, we use a 3-layer DBN with an RBM at each layer. Beside the learning rate and the weight decay in learning the RBMs, which we fix at default values, the only additional parameters are the number of hidden nodes in each layer. In experiments presented here, we use 500 hidden variables in the first and second layer, and 2000 in the third. Changing the "network shape" by modifying these numbers can lead to somewhat different results, giving more exotic combinations, longer ingredient lists, etc.

We learn the network model and then sample the resulting distribution for 10 recipes. A color-map representation of these recipes is shown in Figure 4. In the figure, rows represent the 10 recipes, columns represent 381 ingredient dimensions, and color indicates probability of ingredient presence (blue = unlikely, red = very likely). Ingredients more to the left are those encountered earlier in the dataset, whereas more exotic ones are likely to be present at right. For example, the ingredient most used in the generated recipes, and visible as the connected strip at far left, is *butter*. Recipe contents are given in full in Table 4.5.

A quick inspection of the combinations in the table shows sensible pairings. Ingredients usually used for cakes are combined with fruits or nuts, whereas meat and seafood usually come with vegetables or herbs. Even some more unexpected combinations, e.g. involving fruits and vinegar, or fruits, meat and nuts, are in fact common in African cuisine. Another thing to note is that the number of ingredients is mostly between 10 and 20, out of the possible 381. These lists can be further constrained by increasing the threshold on the ingredient probability given by the network.

5 Conclusions

In this work, we demonstrate some preliminary experiments attempting to learn the ingredient space using machine learning approaches. To that end, we focus only on the ingredient list of each recipe and analyze it in a bag-of-words fashion. Beside exploiting this information to e.g. provide ingredient substitutions or obtain dish similarity networks, we aim to automatically generate novel ingredient combinations. We use Latent Dirichlet Allocation to learn and visualize latent ingredient bases, whereas novel ingredient combinations are generated using Deep Belief Networks. Preliminary results show the promise of this approach, resulting in sensible ingredient bases as well as novel ingredient combinations.

Future work will focus on the gathering of a less biased dataset and on recipe completion under the user-specified constraints. Extensions are likely to include ingredient amounts, as well as better visualization of ingredient and dish relationships.

References

[Ackley *et al.*, 1985] D. H. Ackley, G.E. Hinton, and T.J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[Ahn *et al.*, 2011] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási. Flavor network and the principles of food pairing. *Scientific Reports*, 1(196), 2011.

[Blei *et al.*, 2003] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[Brown *et al.*, 2006] B. Brown, M. Chetty, A. Grimes, and E. Harmon. Reflecting on health: A system for students to monitor diet and exercise. In *Proceedings of CHI Extended Abstracts*, 2006.

[Buykx and Petrie, 2011] L. Buykx and H. Petrie. What cooks need from multimedia and textually enhanced

Table 1: Ingredient lists of newly generated recipes. Ingredients are sorted according to their prevalence in the whole dataset.

recipe	generated ingredient list
1	*egg, wheat, butter, milk, vegetable oil, cream, vanilla, cane molasses, corn, almond, pecan, coconut, tamarind, pineapple, cherry*
2	*egg, butter, onion, garlic, vegetable oil, black pepper, pepper, chicken, lemon juice, mushroom, parmesan cheese, thyme, green bell pepper, white wine, rosemary*
3	*butter, onion, garlic, olive oil, black pepper, vinegar, rice, cheese, parmesan cheese, macaroni, white wine, pea, squash, crab, asparagus*
4	*egg, wheat, butter, milk, vanilla, cinnamon, starch, almond, pecan, raisin, orange peel*
5	*egg, wheat, butter, vegetable oil, cinnamon, walnut, nutmeg, honey, apple, red wine, cranberry, orange peel, cardamom*
6	*wheat, butter, onion, garlic, black pepper, vinegar, carrot, chicken broth, rice, mushroom, soy sauce, lard, lemon, starch, almond, hazelnut*
7	*cream, vanilla, honey, almond, orange, pineapple, gelatin, mint, cranberry, grape juice, raspberry, apricot, rum, orange peel, peach, pear, brandy, plum, pistachio, berry*
8	*egg, wheat, butter, vanilla, cinnamon, walnut, nutmeg, honey, apple, almond, raisin, coconut, oat, nut, cherry, yogurt, cranberry, date*
9	*butter, vanilla, vinegar, cane molasses, lemon, walnut, apple, raisin, cider, fruit*
10	*egg, butter, vanilla, cinnamon, ginger, lemon, nutmeg, almond, coconut, orange, orange juice, apricot, cardamom*

recipes. In *IEEE International Symposium on Multimedia*, 2011.

[Chi *et al.*, 2007] P.-Y. Chi, J.-H. Chen, H.-H. Chu, and B.-Y. Chen. Enabling nutrition-aware cooking in a smart kitchen. In *Proceedings of CHI Extended Abstracts*, 2007.

[Dufour-Lussier *et al.*, 2012] V. Dufour-Lussier, F. Le Ber, J. Lieber, T. Meilender, and E. Nauer. Semi-automatic annotation process for procedural texts: An application on cooking recipes. In *Proceedings of the Cooking with Computers workshop (CwC)*, 2012.

[Gaillard *et al.*, 2012] E. Gaillard, E. Nauer, M. Levefre, and A. Cordier. Extracting generic cooking adaptation knowledge for the taaable case-based reasoning system. In *Proceedings of the Cooking with Computers workshop (CwC)*, 2012.

[Griffiths and Steyvers, 2004] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, volume 101, pages 5228–5235, 2004.

[Hamada *et al.*, 2005] R. Hamada, J. Okabe, I. Ide, S. Satoh, S. Sakai, and H. Tanaka. Cooking navi: Assistant for daily cooking in kitchen. In *Proceedings of the annual ACM international conference on Multimedia*, 2005.

[Hinton *et al.*, 2006] G. E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):15271554, 2006.

[Hofmann, 2001] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[Ide *et al.*, 2010] I. Ide, Y. Shidochi, Y. Nakamura, D. Deguchi, T. Takahashi, and H. Murase. Multimedia supplementation to a cooking recipe text for facilitating its understanding to inexperienced users. In *Proceedings of the Workshop on Multimedia for Cooking and Eating Activities (CEA)*, 2010.

[Mori *et al.*, 2012] S. Mori, T. Sasada, Y. Yamakata, and K. Yoshino. A machine learning approach to recipe text processing. In *Proceedings of the Cooking with Computers workshop (CwC)*, 2012.

[Mota and Agudo, 2012] S. G. Mota and B. D. Agudo. Acook: Recipe adaptation using ontologies, case-based reasoning systems and knowledge discovery. In *Proceedings of the Cooking with Computers workshop (CwC)*, 2012.

[Teng *et al.*, 2011] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic. Recipe recommendation using ingredient networks. In *Proc. 4th International Conference on Web Science*, 2011.

[van der Maaten and Hinton, 2008] L. J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, Nov 2008.

[Wiegand *et al.*, 2012] M. Wiegand, B. Roth, and D. Klakow. Knowledge acquisition with natural language processing in the food domain: Potential and challenges. In *Proceedings of the Cooking with Computers workshop (CwC)*, 2012.