Extracting recipe ingredients from cookbooks

by

Torsten Knauf

A thesis presented for the degree of
Master of Science

Research Group for Communication Systems
at
Faculty of Engineering
Christian-Albrechts-Universität zu Kiel
Germany

31.03.2017

Supervisor: Prof. Dr.-Ing.Norbert Luttenberger
Dr.-Ing. Jesper Zedlitz

Abstract

Always do this one last, when knowing the things to praise yourself for :P

Acknowledgements

If I don't profit from nice people in these thesis, I have done something horrible wrong. So try to remember most of them here at the end... :)

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

A recipe parser, which can tag old and rather unstructured cookbooks according to the ontology of (Schema.org 2016), is developed in this thesis. Once it is tagged, it is easy to extract the tagged entities. The parser is developed and tested with *Davidis, Henriette: Praktisches Kochbuch für die gewöhnliche und feinere Küche. 4. Aufl. Bielefeld, 1849* but can be adapted easily to every German cookbook or website. For other languages new training data and dictionaries for the machine learning preparation of the parser have to be provided, but the general algorithm can be inherited.

This effort is motivated by nutritional science. Being able to extract the ingredients of a recipe automatically simplifies research according healthy food. But also sociological analysis are enabled through that.

The thesis is structured as follows:

Chapter 2

Making a cookbook machine readable

This chapter covers shortly, how we transform a cookbook in a machine readable XML file, which can be arbitrarily processed further. To achieve this, the cookbook has to be digitalized first and afterwards enriched through meta data defined by an ontology.

2.1 Digitalisation

In general there are two different ways, how to digitalize a book. The first one is to scan each side and let an optical character recognition program extract the the text of the scanned picture. The second one is to type it manual into a computer.

The German Text Archive provides a collection of German texts from 16th to 19th century including *Davidis, Henriette: Praktisches Kochbuch für die gewöhnliche und feinere Küche. 4. Aufl. Bielefeld, 1849* in (Deutsches Textarchiv 2016). They digitalized it through double keying, meaning that two people manual typed the book into the computer. Differences in their versions were revised by a third person. They have already enriched the book through *TEI: Text Encoding Initiative*-standard[1]. TEI is a standard for representing real world text in digital form. As many as possible characteristics are kept through meta data. Its main purpose is for analysing in humanities, social sciences or linguistics.

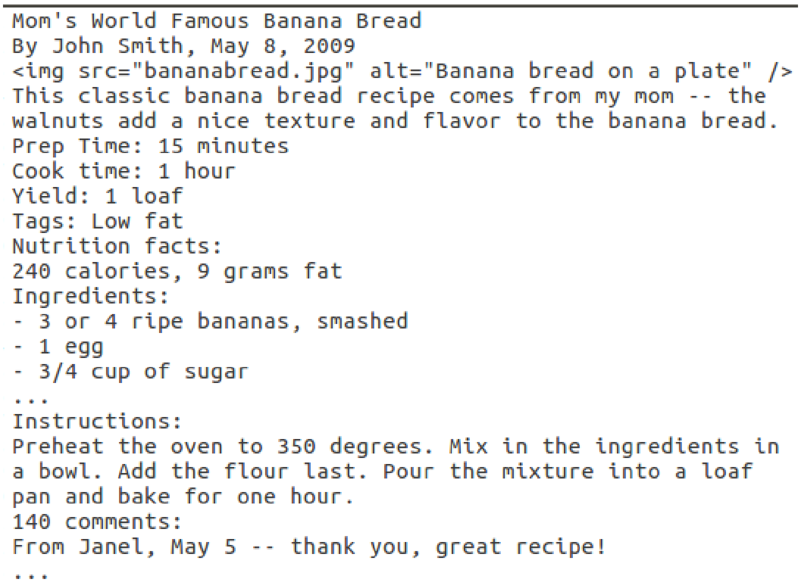
Because we are only interested in extracting certain data from the recipes and not in linguistic analysis or something else, we have transformed the digitalized version as depicted in figure XXX . The essence of this version is, that it now only contains the textual information of the recipes. From this basis we will extract the ingredients as well as their corresponding quantities and units per each recipe.

eispiel Rezepte

2.2 CueML ontology

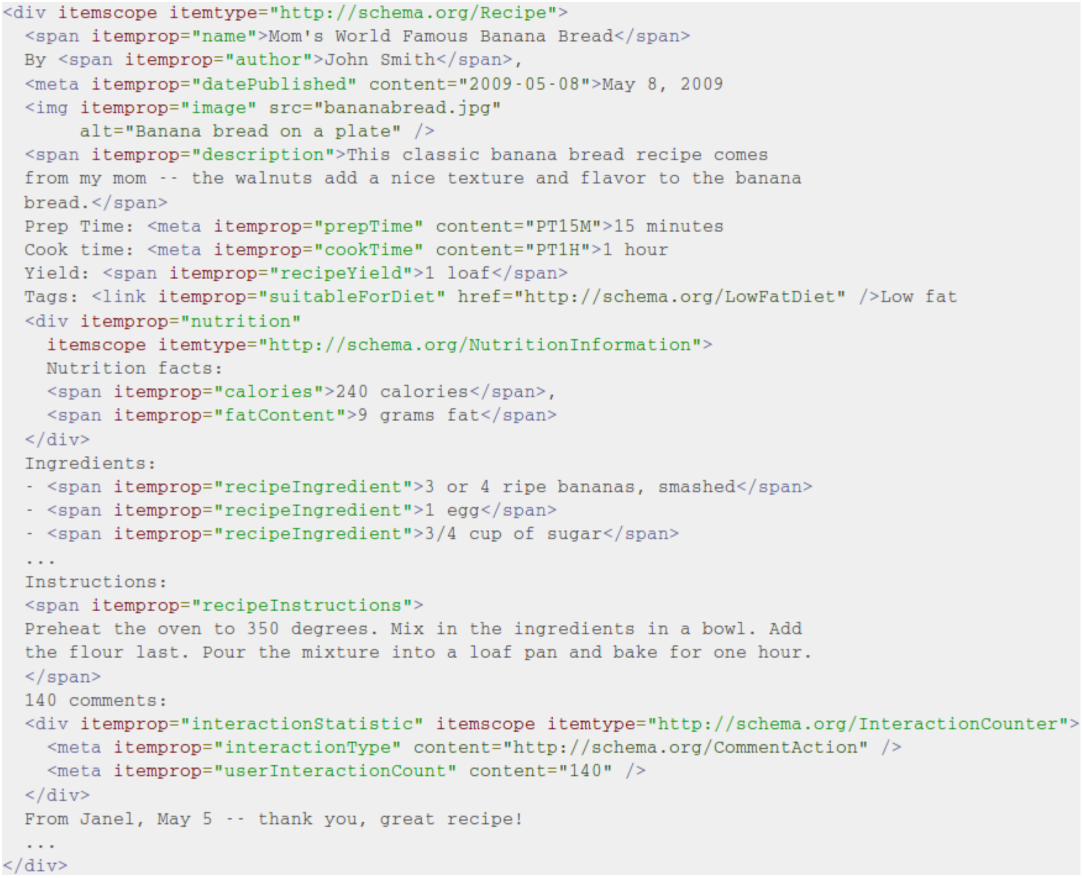
For automatic extraction and further processing of information an ontology is needed. In computer science an ontology is a vocabulary with defined meaning. A description for the use of ontologies, which are a requirement for the Semantic Web, can be found in (Berners-Lee et al. 2000).

[1] http://www.tei-c.org/index.xml



```
Mom's World Famous Banana Bread
By John Smith, May 8, 2009
<img src="bananabread.jpg" alt="Banana bread on a plate" />
This classic banana bread recipe comes from my mom -- the
walnuts add a nice texture and flavor to the banana bread.
Prep Time: 15 minutes
Cook time: 1 hour
Yield: 1 loaf
Tags: Low fat
Nutrition facts:
240 calories, 9 grams fat
Ingredients:
- 3 or 4 ripe bananas, smashed
- 1 egg
- 3/4 cup of sugar
...
Instructions:
Preheat the oven to 350 degrees. Mix in the ingredients in
a bowl. Add the flour last. Pour the mixture into a loaf
pan and bake for one hour.
140 comments:
From Janel, May 5 -- thank you, great recipe!
...
```

(a) A recipe without markup



```
<div itemscope itemtype="http://schema.org/Recipe">
  <span itemprop="name">Mom's World Famous Banana Bread</span>
  By <span itemprop="author">John Smith</span>,
  <meta itemprop="datePublished" content="2009-05-08">May 8, 2009
  <img itemprop="image" src="bananabread.jpg"
       alt="Banana bread on a plate" />
  <span itemprop="description">This classic banana bread recipe comes
  from my mom -- the walnuts add a nice texture and flavor to the banana
  bread.</span>
  Prep Time: <meta itemprop="prepTime" content="PT15M">15 minutes
  Cook time: <meta itemprop="cookTime" content="PT1H">1 hour
  Yield: <span itemprop="recipeYield">1 loaf</span>
  Tags: <link itemprop="suitableForDiet" href="http://schema.org/LowFatDiet" />Low fat
  <div itemprop="nutrition"
    itemscope itemtype="http://schema.org/NutritionInformation">
    Nutrition facts:
    <span itemprop="calories">240 calories</span>,
    <span itemprop="fatContent">9 grams fat</span>
  </div>
  Ingredients:
  - <span itemprop="recipeIngredient">3 or 4 ripe bananas, smashed</span>
  - <span itemprop="recipeIngredient">1 egg</span>
  - <span itemprop="recipeIngredient">3/4 cup of sugar</span>
  ...
  Instructions:
  <span itemprop="recipeInstructions">
  Preheat the oven to 350 degrees. Mix in the ingredients in a bowl. Add
  the flour last. Pour the mixture into a loaf pan and bake for one hour.
  </span>
  140 comments:
  <div itemprop="interactionStatistic" itemscope itemtype="http://schema.org/InteractionCounter">
    <meta itemprop="interactionType" content="http://schema.org/CommentAction" />
    <meta itemprop="userInteractionCount" content="140" />
  </div>
  From Janel, May 5 -- thank you, great recipe!
  ...
</div>
```

(b) The same recipe enriched with markup

Figure 2.1: Schema.org/Recipe example from (Schema.org 2016)

Schema.org/Recipe is an existing ontology for recipes (Schema.org 2016). For example http://cooking.nytimes.com and http://allrecipes.com use it. Its general usage is shown in fig. 2.1 on the previous page. (Food Blogger Pro 2014) is a very nice blog article, which describes the value of it and points out, that its main purpose is to provide help for search engines.

It is not precise enough for general extracting of ingredients. As you can see in fig. 2.1 each line from the list of ingredient is tagged as an ingredient. This is too inaccurate, because the concrete ingredient can still not be understood from the computer and neither its quantity nor unit. Another not suited point is, that the ingredients are only tagged within the list of ingredients, whereby the recipes in our cookbook have no list of ingredients.

Tagging ingredients from the direction texts leads to more complex phrases, which have to be distinguished. We have spotted four different cases, which are shown in fig. 2.2. An additional problem in plain text are cross-references like "prepare dumplings as in previous recipe".

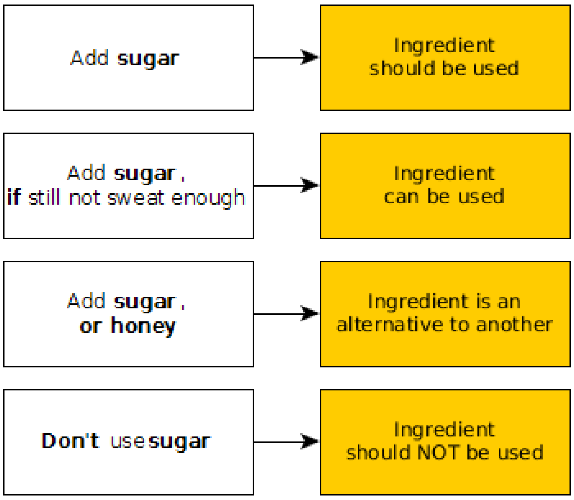

Add sugar
Ingredient should be used
Add sugar, if still not sweat enough
Ingredient can be used
Add sugar, or honey
Ingredient is an alternative to another
Don't use sugar
Ingredient should NOT be used

Figure 2.2: Different meaning of ingredients in direction text

Due to these reasons we came up with **culinary editions markup language (cueML)**. It is pronounced like Kümmel, which is the German word for caraway. First of all we enclose a *recipeIngredient* element more precisely around the ingredient and not vaguely around the whole phrase. Additional we specify an attribute for the *quantity* of an ingredient and another one for its *unit*. Concerning the variants from fig. 2.2 we added attributes which specify if an ingredient *is optional*, an *alternative* or should *not be used*. For the cross-referencing problem we added the attribute *reference*. Its value should point to the referenced element. As a last step we keep the *basis form* of an ingredient in an attribute. That makes it possible to check the ingredient against a dictionary or connect external resources with it, which for example could provide additional nutrition information.

alue of alterna-
ve attri

(a) bla

(b) blubb

Figure 2.3: Example use of cueML

Figure 2.3 shows two example usage of cueML applied to recipes from our targeted cookbook. The full language is described through a RELAX NG grammar, which can be found in appendix B.

Having cueML in place it is easy to extract an ingredient list.

escripe Alg

onverting
ack to
chema.org/Recipe

2.3 Need for automation

As mentioned in (Erdmann et al. 2000), manual tagging is time consuming and error prone. The approach for extracting ingredients from recipes presented in section 3.3 emphasises that. They state, that they need 50-100 fields per recipe. And in the evaluation of their approach they discovered mistakes done in manual tagging.

For tagging one recipe in our cookbook I need about 5 minutes. That means, that I would need more than 2 weeks for tagging only the recipes from our cookbook. Building a big data pool this way is very time consuming and therefore expensive.

Hence automation, which have to be configured only once and can be applied to many resources afterwards, is clearly preferable.

Chapter 3

Related Work

In general there exists many effort about extracting useful information from textual and unstructured resources. The superordinate term for this field of research is Text Mining. It was first mentioned in (Feldman and Dagan 1995) and on overview can be found in (Hotho et al. 2005).

The algorithms for extracting useful information depend highly on existing semi-structures, which can be taken advantage of. Here we present existing algorithm, which we found in the domain of cooking, and distinct their effort from this thesis. But before that, we define precision and recall.

3.1 Precision & Recall

Precision and recall are metrics, which measure the quality of an information extracting algorithm.

$$Precision = \frac{\#(retrieved \cap relevant)}{\#retrieved}, \quad Recall = \frac{\#(retrieved \cap relevant)}{\#relevant} \tag{3.1}$$

They are defined as show in eq. (3.1) according to (Hotho et al. 2005). A high precision states, that the algorithm does only find relevant information as intended. A high recall states, that the algorithm finds many of the total relevant information. Both are needed for an evaluation of an algorithm. If only considering precision, the algorithm could only find the information, which are obvious relevant and therefore find only view information, but having an high score this way. On the other side if only considering recall, the algorithm could return everything. This way its score would have the perfect value of one. But both algorithm are obviously not good, which gets covered by a low value by the other formula.

3.2 Skip The Pizza

(http://skipthepizza.com/ 2012) is a project described on WordPress.org. The author wants to combine his two hobbies cooking and software engineering. For being able to answer questions like "How many ingredients does a typical recipe consist of?" or "Which are the most frequent ingredients?", he extracts the ingredients of recipes from the open source platform http://recipes.wikia.com/wiki/Recipes_Wiki.

Listing 3.1: Shortened example recipe from
http://recipes.wikia.com/wiki/Recipes_Wiki

```
* Makes 6 to 8 servings

== Ingredients ==
* 2 tbsp extra virgin [[olive oil]]
* 3 cloves [[garlic]], finely chopped
[...]

== Directions ==
Heat olive oil and garlic in large skillet over low heat until
garlic begins to sizzle.
Add tomatoes, [...]

[[Category:Cathy’s Recipes]]
[[Category:Garlic Recipes]]
[...]
```

The recipes have a consistent internal representation, which is shown in listing 3.1. The semi-structure, that after `== Ingredients ==` comes a list of ingredients, can be recognized easily. Per line is one ingredient enclosed within `[[ingredient name]]`. Using this semi-structure a regular expression is already good enough for extracting the ingredients from these recipes.

3.3 Extracting Structured Data From Recipes Using Conditional Random Fields

The New York Times (NYT) provides a cooking website with recipes[1]. Their recipes are enriched through Schema.org/Recipes. For providing a recipe recommendation system based on ingredients, you have to extract the exact ingredients from a recipe, which is not enable through this schema as already discussed in section 2.2. Nevertheless they are able to extract them automatically. They use the provided structure from Schema.org/Recipe, that each ingredient phrase from the list of ingredients is enclosed within a *recipeIngredient*-tag, and Conditional Random Fields (CRF) for that. Their approach is described in (The New York Times 2015). So we introduce CRF first and afterwards outline their implementation.

3.3.1 Conditional Random Fields

Given a set of words, CRF wants to predict a suitable set of labels. For example when the set of words is *1 tablespoon salt*, we want to predict *QUANTITY, UNIT, INGREDIENT*, meaning 1 is a quantity, tablespoon an unit and salt an ingredient.

A detailed introduction can be found in (Sutton and McCallum 2012). Here we only want to give a quick overview about linear-chain CRF, because that is the algorithm the NJT uses. Therefore when we write CRF, we mean linear-chain CRF. It is build up from a set of words X, which have already a correct set of labels Y. Such a labelled set is called training data. A joint probability distribution can be extracted from this training data, which states how likely a set of words have a concrete set of labels. Taking the simplified assumption, that each tag depends only

[1] http://cooking.nytimes.com/

on the previous tag and the given set of words, leads to eq. (3.2). This can always be transformed into eq. (3.3). The division with $Z(X, Y)$ ensures, that the value of $p(X, Y)$ is between 0 and 1. $1_{condition}$ is a function which is 1 if the condition is true and 0 otherwise. Smart indexing leads to eq. (3.3). The f_k are called feature functions. The calculation of the Θ_k's is a mathematical optimisation problem. Node that there is very likely no exact solution due to the simplified assumption in eq. (3.2).

$$p(X, Y) = \prod_{t=1}^{T} p(y_t|y_{t-1}) * p(x|y_t), \quad T = \#X \tag{3.2}$$

$$\begin{aligned} p(X, Y) &= \frac{1}{Z(X, Y)} \prod_{t=1}^{T} exp(\sum_{i,j \in S} \Theta_{i,j} * 1_{y_t=i} * 1_{y_{t-1}=j} + \sum_{i \in S} \sum_{j \in O} \mu_{o,i} * 1_{y=i} * 1_{x_t=o}), \\ Z(X, Y) &= \sum_{X} \sum_{Y} \prod_{t=1}^{T} exp(\sum_{i,j \in S} \Theta_{i,j} * 1_{y_t=i} * 1_{y_{t-1}=j} + \sum_{i \in S} \sum_{j \in O} \mu_{o,i} * 1_{y=i} * 1_{x_t=o}), \\ T &= \#X, \quad S = all\ possible\ labels, \quad O = all\ possible\ words \end{aligned} \tag{3.3}$$

$$p(X, Y) = \frac{1}{Z(X, Y)} \prod_{t=1}^{T} exp(\sum_{k=1}^{K} \Theta_k * f_k(y_t, y_{t-1}, X)) \tag{3.3}$$

A join probability distribution can always be transformed into a conditional probability as shown in eq. (3.4).

$$p(Y|X) = \frac{p(X, Y)}{\sum_{Y' \in S} p(Y', X)} \tag{3.4}$$

Having eq. (3.4) in place, a natural prediction function is shown in eq. (3.5), what is exactly what CRF does. The calculation of $prediction(X)$ can be done in $\#S^2 * \#X$through dynamic programming.

$$prediction(X) = argmax_y(P(Y|X)) \tag{3.5}$$

Additional custom feature functions can improve the prediction function. Two example custom feature functions are $f_{k+1}(y_t, y_{t-1}, X) = 1_{X_t\ startswith\ upper\ case}$ or $f_{k+2}(y_t, y_{t-1}, X) = 1_{X_t\ is\ in\ a\ domain\ specific\ dictionary}$.

3.3.2 Implementation of New York Times

feature functions: isCapitalized, inParenthesis, I?, Length of phrases

Eval recall and precision

3.4 Domain Specific Information Extraction for Semantic Annotation

(Ahmed 2009) is a diploma thesis about extracting ingredients and their further processing from recipes. Their algorithm could be divided into two main parts.

First to check every word if it is occurs in a dictionary of ingredients respectively a dictionary of actions and tag it accordingly. For keeping the dictionaries as small as possible they do a morphological Analysis and only store the lemmas of the words. The second main part, and more sophisticated task, is to identify which action should be applied to which ingredient. They have two different approaches for that.

The first one is to do a part of speech tagging and afterwards trying to apply a small set of rules. The example rule 1 means apply the action to all following ingredients and gets matched. Therefore it is extracted that buttermilk and bananas should be extracted.

Listing 3.2: Rule based example

```
add buttermilk and bananas
->
add[ACT] buttermilk[ING] and[CC] bananas[ING]
->
rule 1: VP -> ACT NP (,NP)* (CC NP)?
   -> ACT NP CC NP                      und Hilfsregel:  NP -> DT? JJ* ING
   -> ACT ING CC ING
```

The second one is to do a dependency based parsing, which represent the semantic structure of a sentence in a tree like format.

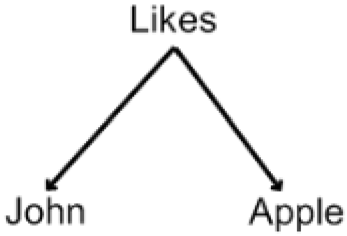


Figure 3.1: John likes apple (Ahmed 2009)

For example fig. 3.1 represents, that the subject of like is John and the liked object is apple. The format as well as building the tree is way more complex than the previous simple rules, but the tree can be build by already existing tools like the Standfod Parser[2]. Having the semantic structure of the sentences, it is trivial to extract which action should be applied to which ingredient.

In their evaluation they apply these two variants to 43 randomly selected recipes from the internet. The precision and recall are presented in table 3.1.

	Precision	Recall
Rule based	97.39%	51.54%
Dependency Based	95.4%	64.12%

Table 3.1: Evaluation Domain Specific Information Extraction for Semantic Annotation

3.5 Distinction to this work

[2] http://nlp.stanford.edu/software/lex-parser.shtml

Chapter 4

Development of recipe parser

4.1 Overall picture

4.1.1 workflow

4.1.2 preparation

4.1.3 evaluation

-Precision -Recall F-measure

4.2 First Iteration: Basis CRF

Git-tag

4.2.1 Idea

4.2.2 Evaluation

4.3 Final recipe parser

4.3.1 Workflow

4.3.2 Evaluation

Chapter 5

Discussion

- Übertragung auf beliebige Bücher (Wenn Buch/Webseite hat bereits Zutatenliste erste Phase entfällt) - TDD / früh Plausibilitäts-Überprüfungen (insbesondere fürs Schemata, Mengenangaben historische Recherche nötig)

5.1 Power of machine readable data

Machine readable data are very powerful. But *with great power comes great responsibility*[1]. In the context of a recipe parser this might be a little bit exaggerated. But specially in mind of the global surveillance disclosures denounced by Edward Snowed with still uncertain dimension, I think it is important to have a consciousness for what can be done. Therefore I want to think about, what can be done through innocent looking machine readable tags.

For the good there exists already much effort for services, which require being able to extract ingredient from recipes.

(e.g. Teng et al. 2012 or Ueda et al. 2011).

Further more, having a huge machine-readable base of recipes and its ingredients, can also provide insights in sociological research. For example in (((Flavor network and the principles of food pairing : Scientific Reports))) is a comparison between American and Asian kitchen based on about 56.000 recipes.

There are many more interesting questions, which could be analysed like a historic analysis of the development and changes of cooking. Occurrences of non-local ingredients or meals are evidence for inter cultural exchange and globalisation. More expensive ingredients could be an indication for prosperity, while very simple kitchen for poverty or even wartimes...

In the bad (The Washington Post 2016) "schwacher vegetarier"

[1] A well known proverb which probably has its origin from the French National Convention during the period of French Revolution. (*With Great Power Comes Great Responsibility* 2015)

Chapter 6

Summary

Appendix A

Statutory Declaration

I declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Master Thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Location, Date Signature

Appendix B

RELAX NG grammar for cueML

Appendix C

Something else else

hello

Bibliography

Ahmed, Zeeshan (2009). "Domain Specific Information Extraction for Semantic Annotation". Diploma thesis. Charles University in Prague, Czech Republic and University of Nancy 2 in Nancy, France.

Berners-Lee, T., J. Hendler, and O. Lassila (2000). "Semantic web". In: *Scientific American, 1(1):68–88.*

Deutsches Textarchiv, Berlin-Brandenburgische Akademie der Wissenschaften (2016). *Davidis, Henriette: Praktisches Kochbuch für die gewöhnliche und feinere Küche. 4. Aufl. Bielefeld, 1849.* URL: `http://www.deutschestextarchiv.de/book/show/davidis_kochbuch_1849` (visited on 10/30/2016).

Erdmann, M. et al. (2000). "From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools". In: *Scientific American, 1(1):68–88.*

Feldman, Ronen and Ido Dagan (1995). "Knowledge Discovery in Textual Databases (KDT)". In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining.* KDD'95. Montréal, Québec, Canada: AAAI Press, pp. 112–117. URL: `http://dl.acm.org/citation.cfm?id=3001335.3001354`.

Food Blogger Pro (2014). *What is recipe schema and how does it impact my food blog?* URL: `https://www.foodbloggerpro.com/blog/article/what-is-recipe-schema/` (visited on 11/06/2016).

Hotho, Andreas, Andreas Nürnberger, and Gerhard Paaß (2005). "A brief survey of text mining". In: *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology.*

http://skipthepizza.com/ (2012). *Skip The Pizza.* URL: `http://skipthepizza.com/blog/analyzing-the-ingredients-of-29200-recipes` (visited on 10/28/2016).

Schema.org (2016). *Schema.org/Recipe.* URL: `https://schema.org/Recipe` (visited on 10/30/2016).

Sutton, Charles and Andrew McCallum (2012). "An Introduction to Conditional Random Fields". In: *Found. Trends Mach. Learn.* 4.4, pp. 267–373. ISSN: 1935-8237. DOI: `10.1561/2200000013`. URL: `http://dx.doi.org/10.1561/2200000013`.

Teng, Chun-Yuen, Yu-Ru Lin, and Lada A. Adamic (2012). "Recipe Recommendation Using Ingredient Networks". In: *Proceedings of the 4th Annual ACM Web Science Conference.* WebSci '12. New York, NY, USA: ACM, pp. 298–307. ISBN: 978-1-4503-1228-8. DOI: `10.1145/2380718.2380757`. URL: `http://doi.acm.org/10.1145/2380718.2380757`.

The New York Times (2015). *Extracting Structured Data From Recipes Using Conditional Random Fields.* URL: `http://open.blogs.nytimes.com/2015/04/09/`

extracting-structured-data-from-recipes-using-conditional-random-fields/?_r=1 (visited on 10/30/2016).

The Washington Post (2016). *Hillary Clinton's health just became a real issue in the presidential campaign.* URL: https://www.washingtonpost.com/news/the-fix/wp/2016/09/11/hillary-clintons-health-just-became-a-real-issue-in-the-presidential-campaign/ (visited on 11/02/2016).

Ueda, Mayumi, Mari Takahata, and Shinsuke Nakajima (2011). "User's Food Preference Extraction for Personalized Cooking Recipe Recommendation". In: *Proceedings of the Second International Conference on Semantic Personalized Information Management: Retrieval and Recommendation - Volume 781.* SPIM'11. Aachen, Germany, Germany: CEUR-WS.org, pp. 98–105. URL: http://dl.acm.org/citation.cfm?id=2887675.2887686.

With Great Power Comes Great Responsibility (2015). URL: http://quoteinvestigator.com/2015/07/23/great-power/ (visited on 11/02/2016).