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... :)

Arbeitsgruppe, stackoverflow, SHK-Dame, Family, yEd

Contents

List of Figures

List of Tables

Chapter 1

Introduction

80-120Seiten anpeilen

1.1   Problem-Stellung

1.2   Mein Beitrag

1.3   Structure of this thesis

The thesis is structured as follows:

Chapter 2

Making a cookbook machine readable

This chapter covers shortly, how a printed cookbook can be transformed into a machine readable format, which can be arbitrarily processed further. To achieve this, the cookbook has to be digitalised first. Afterwards it has to be enriched with meta data from an ontology.

2.1 Digitalisation

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

The German Text Archive provides a collection of German texts from the 16th to the 19th century including our targeted cookbook *Davidis, Henriette: Praktisches Kochbuch für die gewöhnliche und feinere Küche. 4. Aufl. Bielefeld, 1849* in (Deutsches Textarchiv 2016). They digitalised it through double keying, meaning that two people independent of each other manually typed the book into the computer. Differences in their versions were revised by a third person. They have already enriched the book with *TEI: Text Encoding Initiative*-standard[1]. TEI is a standard for representing printed text in digital form. As many as possible characteristics of the printed medium are kept through meta data. Its main purpose is for analysing in humanities, social sciences and 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 digitalised version as depicted in fig. 2.1 on the next page. The essence of this version is, that it is free of for us not relevant information like the encoding of the German ſ and has a clear structure.

---

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

```
<div n="3">
  <head>4. Klare braune Rindflei&#x017F;ch&#x017F;uppe.</head><lb/>
    <p>Die Bereitung die&#x017F;er braunen Kraftbrühe findet man
    in<lb/> <hi rendition="#aq">A.</hi> No. 12. Zu einer
    Ge&#x017F;ell&#x017F;chaft  von 12 Per&#x017F;onen nimmt<lb/>
    man 6 Pfund Rindflei&#x017F;ch und 1 Pfund rohen Schinken. Es
    <lb/> werden braune Klöße No. 3 und Schwammklöße darin gemacht
    .<lb/> Auch kann man nach Belieben braunen Sago darin kochen.
  </p>
</div>
```

(a) Example recipe version from (Deutsches Textarchiv 2016)

```
<cue:recipe type="Suppen." rcp-id="B-4">
  <head>Klare braune Rindfleischsuppe.</head>

  <p>Die Bereitung dieser braunen Kraftbrühe findet man in A. No.
    12. Zu einer Gesellschaft von 12 Personen nimmt man 6 Pfund
    Rindfleisch und 1 Pfund rohen Schinken. Es werden braune Klöße
     No. 3 und Schwammklöße darin gemacht. Auch kann man nach
    Belieben braunen Sago darin kochen.
  </p>
</cue:recipe>
```

(b) Example transformed

Figure 2.1: A recipe from our cookbook

2.2   CueML ontology

An ontology is needed for automatic extraction and further processing of information. In computer science an ontology is a vocabulary with defined meaning. A general description for the use of ontologies can be found in (Berners-Lee et al. 2000).
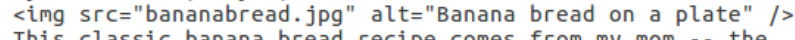
Schema.org/Recipe is an existing ontology for recipes (Schema.org 2016). For example [2] and [3] use it. An example usage is shown in fig. 2.2 on the next page and its main purpose is to support search engines as described in (Food Blogger Pro 2014) and (Schema.org 2016) [4].

But it is not precise enough for further automatic culinary analysis like extracting the ingredients. As you can see in fig. 2.2 each line from the list of ingredients is marked as an ingredient. This is too inaccurate, because the concrete ingredient is not marked and neither its quantity nor unit. Therefore a computer cannot understand it.

---

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

[3]http://allrecipes.com

[4]at http://schema.org/docs/datamodel.html

```
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

```html
<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 2.2: Schema.org/Recipe example from (Schema.org 2016)

That is why we came up with **culinary editions markup language (cueML)**. It is pronounced like Kümmel, which is the German word for caraway. CueML extends TEI, retaining the goals and advantages of TEI this way. Its vocabulary

sticks close to Schema.org/Recipe, so that it profits from the already established schema and can easily be transformed into it, for the sake of search engines. The main refinements are:

- Add attributes for quantity and unit.

- Allow to mark an ingredient as optional through an additional attribute.

- The ability to specify that some ingredients are an alternative to each other.

- Distinction between basic ingredients like salt and complex ingredients like dumplings.

- Comprehend ingredients as unique resources, with possible different form of appearance.

The last point may seem unremarkable but enables a lot. Imagine there would be an freely available set of unique ingredient resources, which specify extra information like its nutrition information and a taxonomy categorisation like is-vegetable. Having the unique resource identifier (URI) of each ingredient from a recipe, its further processing like specifying its fat-level or if it is suitable for a diet becomes trivial, transparent and comparable to other ratings. The only thing you have to do, is collecting the freely available information of your URIs. Unfortunally we could not find such a set of resources. Therefore we used the Bundeslebensmittelschlüssel[5] as first approach, which we could contain with a non-public academic license. The different form of appearance is important, because for example Midder, Kalbsmidder, Kalbsmilch and Bries all describe the same ingredient in German.

A full example is shown on the next two pages. Figure 2.3 is a scan of a recipe from our printed cookbook. In fig. 2.4 is a excerpt, of how we define ingredient resources and fig. 2.5 shows the digital with cueML enriched version of the same recipe.

Schema.org/Recipe is not a well defined grammar. They state "some data is better than none", meaning, that they tolerate wrong meta data for reducing the risk of getting no meta data at all. Furthermore it should be easier to extend the language without an existing grammar. (Schema.org 2016)[6] That is obvious true, because you cannot break, what is not defined. But in contrast to them we defined cueML in a RELAX NG-grammar, which can be found in the attached cd of this thesis. We believe a grammar is a good documentation. In addition a validation against a grammar prevents simple mistakes like typing errors, which can be easily done, when tagging a lot of data.

---

[5]https://www.blsdb.de/

[6]at http://schema.org/docs/datamodel.html

Figure 2.3: Recipe B-16 scanned (Deutsches Textarchiv 2016)

Figure 2.4: excerpt of ingredient elements

```xml
<cue:recipe type="Suppen." rcp-id="B-16">
    <head>Mock Turtle Suppe.</head>

    <p>Es wird hierzu für <cue:recipeYield atLeast="24" atMost="30" unit="people">24–30
        Personen</cue:recipeYield> eine kräftige <cue:recipeIngredient target="#Bouillon"
        >Bouillon</cue:recipeIngredient> von 8–10 Pfund <cue:recipeIngredient
        ref="#Rindkochfleisch" atLeast="8" atMost="10" unit="Pfund"
        >Rindfleisch</cue:recipeIngredient> mit <cue:recipeIngredient ref="#Wurzelwerk"
        >Wurzelwerk</cue:recipeIngredient> gekocht. Zugleich bringt man einen großen
        <cue:recipeIngredient ref="#Kalbskopf" quantity="1"
        >Kalbskopf</cue:recipeIngredient>, eine <cue:recipeIngredient
        ref="#Schweineschnauze" quantity="1">Schweineschnauze</cue:recipeIngredient> und
        <cue:recipeIngredient ref="#Schweineohr">Ohren</cue:recipeIngredient>, einen
        <cue:recipeIngredient ref="#Ochsengaumen" quantity="1"
        >Ochsengaumen</cue:recipeIngredient> und eine geräucherte <cue:recipeIngredient
        ref="#Ochsenzunge" quantity="1">Ochsenzunge</cue:recipeIngredient> zu Feuer und
    kocht dies Alles gahr, aber nicht zu weich. Kalt, schneidet man es in kleine,
    länglich viereckige Stückchen, gibt das Fleisch in die Bouillon, nebst
        <cue:recipeIngredient ref="#braunes_Gewürz">braunem Gewürz</cue:recipeIngredient>,
    ein Paar Messerspitzen <cue:recipeIngredient ref="#Cayennepfeffer"
        quantity="ein Paar" unit="Messerspitze">Cayenne-Pfeffer</cue:recipeIngredient>,
    einige <cue:recipeIngredient ref="#Midder" quantity="einige"
        >Kalbsmidder</cue:recipeIngredient> in Stückchen geschnitten <ref target="#A-16"
        >(siehe Vorbereitungsregeln)</ref>, kleine <cue:recipeIngredient ref="#Saucisse"
        >Saucissen</cue:recipeIngredient>, so viel Kalbskopfbrühe, daß man hinreichend
    Suppe hat, und macht dies mit in <cue:recipeIngredient ref="#Butter"
        >Butter</cue:recipeIngredient> braun gemachtem <cue:recipeIngredient ref="#Mehl"
        >Mehl</cue:recipeIngredient> gebunden. Nachdem dies Alles ¼ Stunde gekocht hat,
    kommen noch <cue:recipeIngredient target="#L-4">Klöße von
        Kalbfleisch</cue:recipeIngredient>, einige hart gekochte <cue:recipeIngredient
        ref="#Ei" quantity="einige">Eier</cue:recipeIngredient> in Würfel geschnitten, ein
    Paar Eßlöffel <cue:recipeIngredient ref="#Englische_Soja" quantity="ein Paar"
        unit="EL">Engl. Soja</cue:recipeIngredient> hinzu, und wenn die Klößchen einige
    Minuten gekocht haben, ½ Flasche <cue:recipeIngredient ref="#Madeira" quantity="0.5"
        unit="Flasche">Madeira</cue:recipeIngredient> und auch <cue:recipeIngredient
        ref="#Auster" optional="True">Austern</cue:recipeIngredient>, wenn man sie haben
    kann. Dann wird die Suppe sogleich angerichtet.</p>

    <note>Anmerk. Der <cue:recipeIngredient ref="#Englische_Soja" optional="True"
        >Soja</cue:recipeIngredient> macht die Suppe gewürzreicher, kann jedoch gut
    wegbleiben, und statt <cue:recipeIngredient ref="#Madeira" altGrp="1"
        >Madeira</cue:recipeIngredient> kann man <cue:recipeIngredient
        ref="weißen_Franzwein" altGrp="2">weißen Franzwein</cue:recipeIngredient> und
    etwas <cue:recipeIngredient ref="#Rum" altGrp="2" quantity="etwas"
        >Rum</cue:recipeIngredient> nehmen<cue:alt target="1 2"/>. Sowohl die
        <cue:recipeIngredient ref="#Boullion">Bouillon</cue:recipeIngredient> als
        <cue:recipeIngredient ref="#Kalbskopf" quantity="1"
        >Kalbskopf</cue:recipeIngredient> können schon am vorhergehenden Tage, ohne
    Nachtheil der Suppe, gekocht werden.</note>
</cue:recipe>
```

Figure 2.5: Recipe B-16 enriched with cueML

2.3   Need for automation

The tagging requires some domain knowledge. For example we first assumed that the ingredient "Scorzoner Wurzel" is burgundy truffles. But when we realized, that it should be put into many soups, we started to doubt that. We now reason, that "Scorzonera hispanica" is meant, which is the name of salsify. Another example is bouillon. Today we would simple buy bouillon powder in the super market and mix it with water. But at the time of our cookbook it was common to cook one from the scratch and therefore there is an instruction for bouillon. Beside the cooking domain knowledge also some XML comprehension is needed for the tagging.

For tagging one recipe in our cookbook I need about 5 minutes in average. That means, that I would need more than 2 weeks of full time work, for tagging about 1000 recipes. We had a student assistant, who has never worked with XML before. In about 18 hours she could tag 66 recipes, meaning she needed more than 15 minutes in average per recipe. To be fair, her average speed would be probably better now, after she got used to XML.

Beside the needed domain knowledge and time effort, the tagging is also error prone. Looking over the tagged recipes, I found mistakes, some done by me and some more done by our student assistant.

That tagging is time consuming and error prone get emphasised by (Erdmann et al. 2001) as well as by the New York Times's approach to extract ingredients from recipes, which we will present in section 3.2.

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

Chapter 3

Related Work

In general there exist many works 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 an 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 algorithms, which we found in the domain of cooking, and distinct their effort from this thesis.

3.1   Skip The Pizza

(*Skip The Pizza* 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.

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. The quantities and units of ingredients are not of interest for this work.

```
* 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]]
[...]
```

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

3.2 Extracting Structured Data From Recipes Using Conditional Random Fields

The New York Times provides a cooking website with recipes[1]. Their recipes are enriched with 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 enabled 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 marked as recipeIngredient, and Conditional Random Fields (CRF) for that. Their approach is described in (The New York Times 2015). Hence we introduce here CRF first and afterwards outline their implementation.

3.2.1 Conditional Random Fields

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

A detailed introduction to CRF 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 New York Times uses. Therefore, when we write CRF, we mean linear-chain CRF.

Its starting point are vectors of words X, which have already correct vectors of labels Y. Such related sets of vectors are called training data. A joint probability distribution can be extracted from this training data, which states how likely a vector of words has a corresponding vector of labels. Taking the simplified assumption, that each label depends only on the previous label and that the current word depends only on the current label, leads to eq. (3.1). This can always be transformed into the form of eq. (3.2). The division with $Z(X, Y)$ ensures, that the value of $p(X, Y)$ is between zero and one. $1_{condition}$ is a function, which is one if the condition is true and zero otherwise. Smart indexing leads to eq. (3.3). Each f_k is called a feature function. The calculation of the Θ_k's is a mathematical optimisation problem. Note that there is very likely no exact solution due to the simplified assumption of eq. (3.1).

$$p(X, Y) = \prod_{t=1}^{T} p(y_t | y_{t-1}) * p(x_t | y_t), \quad T = \#X \tag{3.1}$$

---

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

$$p(X,Y) = \frac{1}{Z(X,Y)} \prod_{t=1}^{T} exp(\sum_{i,j \in S} \Theta_{i,j} * 1_{y_t=i} * 1_{y_{t-1}=j} + \sum_{i \in S} \sum_{j \in O} \mu_{o,i} * 1_{y=i} * 1_{x_t=o}),$$

$$Z(X,Y) = \sum_{X} \sum_{Y} \prod_{t=1}^{T} exp(\sum_{i,j \in S} \Theta_{i,j} * 1_{y_t=i} * 1_{y_{t-1}=j} + \sum_{i \in S} \sum_{j \in O} \mu_{o,i} * 1_{y=i} * 1_{x_t=o}),$$

$$S = all\ possible\ labels, \quad O = all\ possible\ words$$

$$(3.2)$$

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

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

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

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

The described model so far is a Hidden Markov Model. In a CRF you are also allowed to take only a subset of these features, what can improve performance without loosing accuracy. Additional improvement in a CRF can very likely be achieved by further custom feature functions, which may depend on the whole vector of words, instead of only the current word identity. For example some custom feature functions could be:

- $f_{k+1}(y_t, y_{t-1}, X) = 1_{x_t\ startswith\ upper\ case}$

- $f_{k+2}(y_t, y_{t-1}, X) = 1_{x_t\ is\ in\ a\ domain\ specific\ dictionary}$

- $f_{k+3}(y_t, y_{t-1}, X) = 1_{x_t\ is\ in\ a\ domain\ specific\ dictionary\ and\ x_{t-1}\ is\ an\ article}$

Putting all together leads to definition 3.1. Note that after plugging eq. (3.3) into eq. (3.4) Z now only depends on X.

Definition 3.1 (linear-chain Conditional Random Field) *A linear-chain Conditional Random Field is a conditional distribution of the form:*

$p(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{T} exp(\sum_{k=1}^{K} \Theta_k * f_k(y_t, y_{t-1}, X)),$ *where*
    $Z(X) = \sum_S \prod_{t=1}^{T} exp(\sum_{k=1}^{K} \Theta_k * f_k(y_t, y_{t-1}, X))$ *and*
    Y vector of labels, X vector of words, S set of all possible labels

---

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

$$prediction(X) = argmax_Y(P(Y|X)) \qquad (3.5)$$

3.2.2   Implementation of New York Times

As starting point the New York Times have manually specified labels for over 130,000 ingredient phrases, which they use as training data. They are extracted out of their website, where their recipes are already enriched with Schema.org/Recipe.

One example ingredient phrase is shown in listing 3.2. They use CRF++[2] as library, which is an implementation of CRF. The first word per column is the actual word. The last word is the label, which should be predicted in IOB2 format. The format states, that the beginning of an entity gets prefixed with *B-* and the continuation with *I-*, as you can see in the last three words. Words, which do not belong to a relevant entity, get labelled with *OTHER*. The labels in between are custom features.

```
3/4        I1   L12   NoCAP   NoPAREN   B–QTY
pound      I2   L12   NoCAP   NoPAREN   OTHER
shiitake   I3   L12   NoCAP   NoPAREN   B–NAME
mushrooms  I4   L12   NoCAP   NoPAREN   I–NAME
,          I5   L12   NoCAP   NoPAREN   OTHER
stemmed    I6   L12   NoCAP   NoPAREN   B–COMMENT
and        I7   L12   NoCAP   NoPAREN   I–COMMENT
quartered  I8   L12   NoCAP   NoPAREN   I–COMMENT
```

Listing 3.2: Extract of the training data for New York Times CRF

The feature functions, which the CRF should use, are specified through the templates of listing 3.3. They have the format *U00:%x[row,column]*. The row specifies the targeted word in relative position to itself and the column specifies the absolute position. Therefore when the actual word is mushrooms the data from listing 3.2 would be expended to the values in listing 3.4.

---

[2]`https://taku910.github.io/crfpp/`

```
# Unigram                          pound
U00:%x[-2,0]                       shiitake
U01:%x[-1,0]                       mushrooms
U02:%x[0,0]                        ,
U03:%x[1,0]                        stemmed
U04:%x[2,0]                        I4
U05:%x[0,1]                        L12
U06:%x[0,2]                        NoCAP
U07:%x[0,3]

U08:%x[-2,4]                       NoPAREN
U09:%x[-1,4]                       NoPAREN
U10:%x[0,4]                        NoPAREN
U11:%x[1,4]                        NoPAREN
U12:%x[2,4]                        NoPAREN

U13:%x[0,0]/%x[0,2]                mushrooms/L12
U14:%x[0,1]/%x[0,2]                I4/L12
U15:%x[0,0]/%x[0,3]                mushrooms/NoCAP
U16:%x[0,0]/%x[0,4]                mushrooms/NoPAREN
U17:%x[0,0]/%x[0,1]                mushrooms/I4

# Bigram
B                                  I-NAME  B-Name
```

Listing 3.3: Feature templates for New York Times CRF

Listing 3.4: Derived values when the current word in listing 3.2 is mushrooms

Feature functions are extracted from these templates and the corresponding values of the training data as shown in listing 3.5. The B-template takes the transition from y_{t-1} to y_t into account. Note that for example the custom feature function, if a word is inside parentheses, gets implemented through adding a label YesPAREN/NoPAREN to the trainings data in the fourth column and specifying the feature template U10%[0,4]. Also note, that for each such template $\#(all\ possible\ labels) * \#(all\ possible\ words)$ different feature functions gets created. For the B-template $\#(all\ possible\ labels)^2 * \#(all\ possible\ words)$ different feature functions gets created.

```
U02:%[0,0]              → 1_{y_t=I-Name and x_t=mushrooms}
U02:%[-2,4]             → 1_{y_t=I-Name and x_{t-2} 4th label isNoPAREN}
U13:%x[0,0]/%x[0,2]     → 1_{y_t=I-Name and x_t=mushrooms and x_t 2nd label isL12}


B
                        → 1_{y_t=A and y_{t-1}=B} for each A, B ∈ possible prediction labels
```

Listing 3.5: Extracting of feature functions from templates

Hence the only custom features the New York Times take into account for prediction are the word identity (column 0), the index of the current word (column 1), the length of the sentence (column 2, rounded to 4, 8, 12, 16 or 20), if the word

starts capitalized (column 3) and if the word is inside parenthesis (column 4), in relation with the current word and some surrounding words.

When applying their algorithm to 481 example recipes, they get 89% sentence-level accuracy, meaning that they get 9 out of 10 ingredient phrases completely right. They tested only so few recipes, although they had over 130,000 ready labeled recipes, because they found no way for automatic evaluation. Their problem was, that there were to many ambiguous phrases and to many mistakes in the manually labeled data.

3.3 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 in a dictionary of ingredients respectively a dictionary of actions and label 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 a rule based approach and a dependency parsing based approach for that.

The rule based approach is to do a part of speech tagging (POS) first and afterwards try to apply a small set of rules, which define a context free grammar, with predefined meaning. Listing 3.6 shows an example, meaning buttermilk and bananas should be added.

```
# Example sentence
add buttermilk and bananas

# Example sentence with POS and dictionary-label
add[ACT] buttermilk[ING] and[CC] bananas[ING]

# Apply rule VP -> ACT NP (,NP)* (CC NP)?
# Predefined meaning of the rule: Apply ACT to all ING
-> ACT NP CC NP
# Apply rule NP -> DT? JJ* ING twice
-> ACT ING CC ING

# With:
# VP,NP ∈ non-terminal symbols, CC ∈ Conjunctions, DT ∈ Determiner,
    ACT := action, ING := ingredient
```

Listing 3.6: Rule based example

The second approach is to do a dependency based parsing, which represents the semantic structure of a sentence in a tree like format. 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 Stanford Parser[3]. The semantic structure of the sentences give strong evidence, which action should be applied to which ingredient.

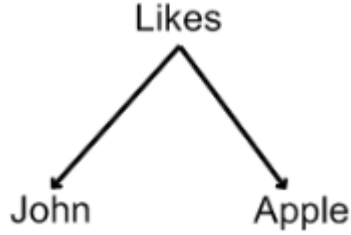[3]http://nlp.stanford.edu/software/lex-parser.shtml

Figure 3.1: John likes apple (Ahmed 2009)

In their evaluation they apply these two variants to 43 randomly selected recipes from the internet. Their precision and recall are presented in table 3.1. Even the more sophisticated dependency based approach has only a recall of 64%. This is due to a lack of mapping actions to their corresponding ingredients.

	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

3.4 Data-driven Knowledge Extraction for the Food Domain

(Wiegand et al. 2012) are interested in relations of ingredients. They analyse the four relations:

1. INGREDIENT typically suits to EVENT

2. INGREDIENT is typically served with other INGREDIENT

3. INGREDIENT can be replaced with other INGREDIENT

4. INGREDIENT is part of RECIPE

Their data set consists of pure text, divided into title and body. They use GermaNet[4] for the extraction of ingredients. GermaNet is a thesaurus for German words providing a category "Nahrung", which means food. When the lemma of a word appears in this category, the word is assumed to be an ingredient. They have three for us relevant methods for instantiating their relations. We only cover their connection the relation *Ingredient is part of recipe*, because that is the one we are interested in.

Title and body instantiates for each ingredient of a body a relation, if the title is a recipe. This way they reach a P@10 of 0.776, meaning that when ask for the ingredients of a recipe almost 8 of 10 are right. We want to mention, that this value is a lower boundary. For example when a recipe consists only of 5 ingredients, the maximal possible P@10 value would be 0.5.

---

[4]http://www.sfs.uni-tuebingen.de/GermaNet/

The other two methods are **Pattern-based** and **Co-occurrence**. Both evaluation are very bad and therefore we only discuss the idea of them. Pattern-based searches for patterns like *RECIPE made of INGREDIENT*. They state, that when a pattern is matched, it is always a good match, but that they match not often. They also state, that patterns, based on dependency based parsing, like in the previous section, are not an improvement, due to domain specific mismatch of the parser. Co-occurrence looks for co-occurrences in the text of a recipe and an ingredient.

Last they note, that the relations 2 and 3 are often mixed-up, because they are very similar to each other.

3.5   Lessons for this work

(*Skip The Pizza* 2012) exploits a strict semi-structure, which is not given in our very unstructured cookbook. Therefore we need more advanced algorithm than regular expressions.

(The New York Times 2015) take advantage of the point, that they have already marked the ingredient phrases from the list of ingredients. But our cookbook has no list of ingredients. Nevertheless, the ingredient phrases could be extracted from plain text through a CRF algorithm first and afterwards we could use the algorithm of (The New York Times 2015). Another option would be to just try a CRF on the plain text guessing all the work at once. However we have to consider, that we do not have near as 130,000 training data. Our lack of training data could be caught up by more sophisticated custom feature functions.

(Ahmed 2009) only uses plain text. But they have a bad recall, due to the point, that they are not good in mapping actions to their corresponding ingredients. In contrast to mapping actions to ingredients, we want to map quantities to units and these in turn to ingredients, which could be an easier task. Anyway, the dictionary check, enabled by a morphological analysis, from their algorithm first main part, could be a good custom feature function for a CRF.

(Wiegand et al. 2012) is interesting, because we target recipes and therefore can easily do their Title and body method. They also enforce (Ahmed 2009), that pattern-based extraction is good, when the patterns match, what is unfortunately not often the case. The Co-occurrence method is an interesting idea for categorization, if an ingredient is optional, or an alternative to another one. For example the co-occurrence of *two ingredients* and the word *instead* within one sentence indicate, that the two ingredients are an alternative to each other.

Chapter 4

CRF-based extraction

We present a very basic prototype for extracting ingredients from recipes based on CRF in this chapter.

4.1   CRF prototype

This prototype can be found under the following git tag[1]. It uses the SWIG API of crfsuite[2] as python binding. For labeling we take the tags `B-Quantity`, `I-Quantity`, `B-Unit`, `B-Ingredient`, `I-Ingredient`, `B-CookTime`, `I-CookTime`, `B-Yield`, `I-Yield`, `B-IngredientYield`, `I-IngredientYield` `B-Link`, `I-Link` and `O` for others. We tag 9 of our recipes with these tags. For illustrating the beginning of such a tagged recipe is shown in listing 4.1. Our used features are word identity and label transitions.

```
#B−1
Klare             O
weiße             O
Rindfleischsuppe  O

Man               O
rechnet           O
,                 O
wenn              O
die               O
Gesellschaft      O
klein             O
ist               O
,                 O
auf               O
jede              O
Person            O
3/4               B−Quantity
Pfund             B−Unit
,                 O
```

---

[1] https://github.com/Torsten89/SourceCodeExtractingRecipeIngredientsFromCookbooks/tree/FirstCRFPrototype

[2] http://www.chokkan.org/software/crfsuite/

```
bei                 O
einer               O
großen              O
Gesellschaft        O
1/2                 B−Quantity
Pfund               B−Unit
Fleisch             B−Ingredient
.                   O


...
```

4.2 Evaluation of this prototype

Although this is only a simple prototype, it raises many questions.

First we do a little test on a recipe from our cookbook, which was not in the training data. It can be found in **??**. It consists of 96 words. From these 96 words the CRF got all 83 `O`-tags correctly. From the 13 tags other than `O` it missed the 5 shown in listing 4.2. *Eine* and *einige* means *some* and is a vague quantity. The other three are potatoes and potato dumplings.

```
eine                B−Quantity      O
einige              B−Quantity      O
Kartoffeln          B−Ingredient    O
Kartoffeln          B−Ingredient    O
Kartoffel−Klöße     B−Ingredient    O
```

Listing 4.2: Differences

Eine and *einige* appear very often in German and are only sometimes the quantity of an ingredient. Therefore it is hard for a CRF to get this right. But because the quantity is vague, it is questionable to mark it anyway.

Potatoes could be likely caught through a custom feature function $1_{word\ is\ in\ dictionary\ of\ ingredients}$. But having that in place, one has to ask for the advantage of the CRF, when we can already extract the ingredients with a dictionary.

We decided to not follow up the CRF approach, because we bothered with following limitations, as we reasoned about the CRF:

- Further feature and label engineering is not straight forward. First of all we have only considered ingredients. For optional ingredients as well as for alternative ingredients two new labels are required. Also sometimes a recipe introductions tells you to not use an ingredient, which could be marked with an own label. But these four labels are very similar, and therefore the CRF probably needs more features to distinguish between them. One such a feature could be, if *don't* is within the sentence. (Time consuming!)

- A CRF is not able to extract relations between entities. Once the ingredients, quantities and units are extracted, they still have to be matched together.

Also when ingredients are marked as alternative ingredients, it is still unclear which one are an alternative to which.

- Only labeling ingredients is a simplification of our cueML. We would also like to distinguish between basic and complex ingredients. When it is a basic ingredient, we would like to add a reference to the Bundeslebensmittelschlüssel. And when it is a complex ingredient, we would like to add a reference to a recipe of it. Also breaking down, what kind of meat, when a sentence only mentions meat, is not possiblle within the CRF.

- Above all a further development is to time consuming for this master thesis. The 9 trainings recipes lead to more than 1400 lines, which have to be tagged manually. Assuming 10 seconds per line, that would take more than three hours. 10 seconds may seem pessimistic, but consider the following translated example from our cookbook: *Take the veal as in number one, according to the number of people take a little bit more, because it has less taste than beef.* When not reading carefully, it is easy to tag *beef* as an ingredient. And as mentioned before we not only need to tag it once, but probably multiple times, because our labels and features are not clear yet.

  The New York Times had more than 130,000 manual labeled trainings recipes. The CoNLL-task[3], what is a common reference for CRF tasks, provides more than 10,000 labeled sentences. Building such huge training sets is out of scope for this project.

---

[3](Tjong Kim Sang and De Meulder 2003)

Chapter 5

Dictionary- and Rule-based extraction

First we introduce here a basic prototype, which is based upon dictionary extraction and further rule-based processing. After that we continue to develop this approach, because we find it promising.

5.1 Dictionary- and Rule-based prototype

-lemmas mit -wurzel aufpimpung

5.1.1 Evaluation

- besser testbar (testcase vs crf neue features -> funktioniert alles noch, anderes funktioniert vll besser)

5.2 Weitere Rules

Für altGrp, optional, dontUse

5.3 Application to recipes from Chefkoch.de

Auf Introduction Text / Zutatenliste?

5.4 GermaNet

5.5 Evaluation

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

$$Precision = \frac{\#(retrieved \cap relevant)}{\#retrieved}, \quad Recall = \frac{\#(retrieved \cap relevant)}{\#relevant} \quad (5.1)$$
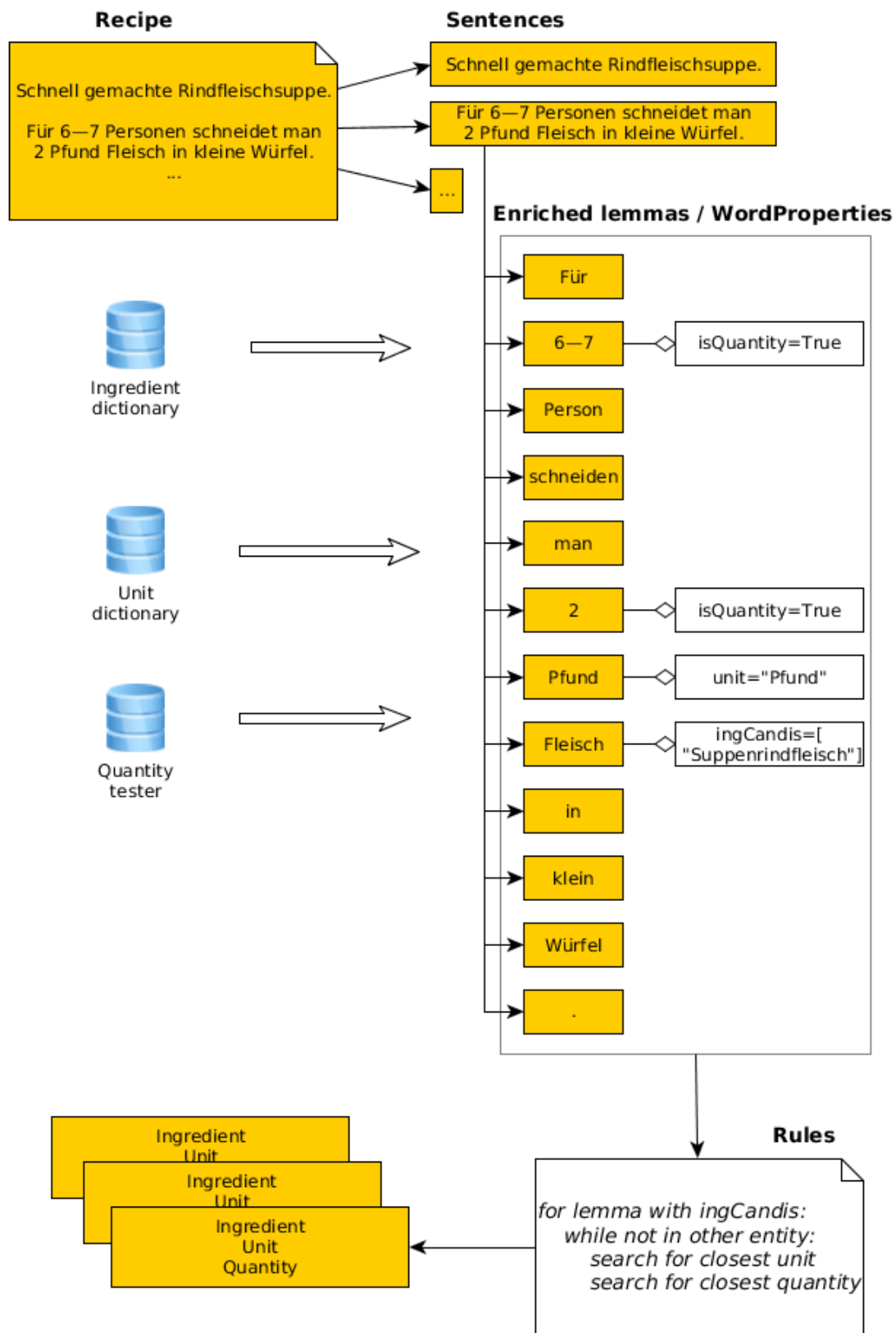
Figure 5.1: Dictionary- and Rule-based workflow

They are defined as shown in eq. (5.1) according to (Hotho et al. 2005). A high precision states, that the algorithm does only find relevant information as intended. The ideal precision of one would mean, that all extracted data where useful. A high recall states, that the algorithm finds many of the total relevant information. The perfect score of one would mean, that all relevant information were found. Both are needed for the 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 a 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 obvious not good, which gets covered by a low value of the other formula.

Chapter 6

Discussion

6.1   Usefulness of automatic extraction of ingredients

I personally feel it is an interesting computer science challenge to extract ingredients from plain text and make them machine readable. But that would be a bad justification for a master thesis. This section points out some of many applications, which are possible, once the ingredients are extracted. Building a database manual for all these applications is way to time consuming, as pointed out in 2.3. Therefore this thesis is useful, what is a good motivation as well as a good justification for this master thesis.

(Ueda et al. 2011) build a **recipe recommendation system** for a cooking website, based on which ingredients are liked by the users and which not.

Beside that, there seems to exist a lot of effort about **diet, targeting the well being**. (Teng et al. 2012) analyse, how far a recipe is liked in relation with its nutrition information, which are based on the ingredient of the recipe. They also analyse how far an ingredient can be replaced through another ingredient, which is nice, if you want a low-fat alternative, or it is not available right now. (Freyne and Berkovsky 2010) as well as (Geleijnse et al. 2010) build a recipe recommendation, which prefer healthy recommendations. As before, the healthiness of a recipe is determined through its used ingredients.

Both, a recommendation system as well as help with healthy diet, do obviously increase the attractiveness of a cooking website. That raises the question, how much are these features worth. Unfortunally it is not an easy task to determine the value of cooking websites. Chefkoch.de, which is the biggest German cooking website, is a 100% subsidiary of Gruner+Jahr today, but there are no reliable information about the cost of acquisition. They are also not allowed to share any information about their revenue, but have more than 100 employees, what is an indication for a lot of revenue (see appendix B). Therefore these application are probably also worth a lot.

Social science is also possible through analysing recipes. For example (Henning 2008) analyses typical characteristics of cookbook through German history. She noticed that recipes before the 18.th century often contained ingredients, which should mask the taste of rotted food, what is clearly a sign for poverty and food shortage. Fasting meals covered about a third of many cookbooks, which emphasises the influence of the church at that time. First mentions of non-native ingredients and meals are indications for intercultural exchange. Rare and expensive ingredients are

signs for prosperity while possible substitution of them for poverty. Another example is in (Ahn et al. 2011). They compare American and Asian culinary practice based on about 56.000 recipes.

6.2 Contribution to field of research

- CueML - validation durch prog da mensch nicht einheitlich tagt (Brühe mal und mal nicht zutat) - rule-based nicht exakt match sondern co-occurrence von Indikationswörtern (Ingredients und "anstatt") - Übertragung auf beliebige / allgemeine Rezepte -CRF (co-reference von wörtern, pos etc automatisch (wie groß/langsam wird das Model?), aber eigentlich nicht der richtige Alg) vs Dict based (gut genug, besser testbar, auf andere Sprachen überetztbar in dem nur Regel überetzten und nicht neue Trainingsdaten (wie viele werden überhaupt benötigt?))

6.3 Quality of the extracted data

- Mengenangaben ohne recipeYield witzlos - Übersetzung von (historischen) Mengenangaben (Maß) - ganzes Gewürz - Ungenaue Mengenangaben (ein Stich, 5Pfennig Brot, etwas) - links/refs erschweren Auswertung (koche Nr. 7 vs wie in Nr. 7) - Nährwertangaben bei Gerichten wo target="L-4" -In B-2 "Anmerk. Will man Reis oder Sago zur Suppe nehmen, so gibt man dieses später hinein. Man rechnet davon auf jede Person bei allen Fleischsuppen einen gestrichenen Eßlöffel voll." - Wichtigkeit der Basisformen (Cayennepfeppfer entdeckt nicht Cayenne-Pfeffer) Infos aus Allg. Vorteil eines Kapitels (Reis 15g pro Suppe)

-Mensch kann Fehler erkennen (B-10 Rindfleisch ->Hasen /falscher Vreweis)

- BLS Angaben für alk. freie Getränke :D :D :D, nicht open source, nicht word wide - welche Zutat welche BLS key (BLS überraschend schwer :D)

- Publishen / Sharen von cueML - (Ahmed 2009) also points out that ontology building is time consuming and expensive - Schema alignment paper - nicht ein schema für alle Zwecke

6.4 Developing

- Bachelorarbeit prototypen die out of hand gingen... - Prototype, Metapher der 2 Hüte (neues feature und refaktorisieren), tests

- TDD / früh Plausibilitäts-Überprüfungen - Tagen als Hi-Wi war ne dumme Aktion'(insbesondere fürs Schemata, Mengenangaben und Namen historische Recherche nötig (Gefühl für die Daten, Wurzelwerk, braunes Gewürz, Engl. Soja, tagging schema ist schwieriger als gedacht)) - CRF zu aufwändig abre vll gut wegen co-occurence of indicator words automatisch statt manuelle rules (die abre auch nicht so schlimm sindi

- wenn weitere tags (Links, cooktime, yield, ...) xmlElems (kind, start, end) statt ingsOfSentence = [], linksOfSentence = []..., gleiches mit if else in getTokensWithExtractedIngs (ref Bachelorarbeit, smell)

6.5 Knowledge is power

The goal of this thesis is to facilitate some machine readable data and machine readable data provides a lot of knowledge. The following has not the claim to be a scientific research. It should provide some critical thoughts and sensitize for the power of data. From my point of view it should be discussed controversially, because that would strengthen the awareness of this topic.

Already in the seventh-century Imam Ali said *"Knowledge is power and it can command obedience. A man of knowledge during his lifetime can make people obey and follow him and he is praised and venerated after his death. Remember that knowledge is a ruler and wealth is its subject"*[1]. How this thesis can support wealth was already discussed in section 6.1.

But **with great power comes great responsibility**[2]. I want to give three examples, where knowledge, provided through data, is powerful and could be abused:

Article 136 from the constitution of the Weimar Republic from 1919 gave the state the right to acquire the religion affiliation of their citizen in a census. That makes sense, because the state does collect the church taxes. 1939 the regime of Adolf Hitler used that right in a census, to force everybody to tell if themselves, one of their parents or one of there grandparents are Jews. The result lead to a categorization of full Jew, hybrid of level one or hybrid of level two. It is uncertain today, how far the regime accessed and used these data[3]. In my opinion it is likely, that they did take use of these information in some way. Otherwise they would not have collected it. But it is undoubted, that the knowledge, if your family are Jews, could lead to deportation into a concentration camp.

Second the question how far *Fake News* influence elections is actual very present in the media[4]. News are also data and many citizen take their political knowledge from them. A person, who can change arbitrarily opinions of people before an election, is clearly powerful.

Third, also very present in the media, is the claim, that Russia has collected material to blackmail the American president Donald Trump[5]. This ability is clearly powerful, too. I personally would be surprised, if they had not done that, because I accuse western countries to do the same to some point. The global surveillance disclosures revealed that the American and British secret services monitored huge parts of the internet. That is certainly expensive. I assume, that one payback is the hope to find material, which they can use for their political advantage. For example the mobile phone of the German chancellor was monitored since 2002[6]. It takes not much imagination to associate that with the declining of the Iraq War of the actual chancellor as well as with his closeness to Russia.

---

[1]Imam Ali, Nahj Al-Balagha, Saying 146

[2]A well known proverb which probably has its origin from the French National Convention during the period of French Revolution. (`http://quoteinvestigator.com/2015/07/23/great-power/` visited on 11/02/16)

[3]`https://de.wikipedia.org/wiki/Volkszählung_im_Deutschen_Reich_1939#Kontroverse_Deutungen` visted on 28/12/16

[4]E.g `https://www.nytimes.com/2016/11/18/technology/fake-news-on-facebook-in-foreign-elections-thats-not-new.html?_r=0` visted on 12/11/16

[5]E.g.`https://www.nytimes.com/2017/01/10/us/politics/donald-trump-russia-intelligence.html` visted on 10/01/17

[6]`http://www.spiegel.de/politik/deutschland/handy-der-kanzlerin-die-wichtigsten-fakten-der-abhoeraffaere-a-930411.html` visited on 11/01/17

In the context of a program, which makes recipes machine readable, this all might be total exaggerated. But the health of Hillary Clinton was a topic at the American president election 2016[7]. I personal know, that you should not eat some ingredients, when you have pancreatic cancer. It probably would have had effected survey results, when someone had tracked Hillary's internet traffic and stated Hillary had pancreatic cancer, based on the fact, that she was interested in recipes, avoiding these ingredients. I also wonder, if revealing Hillary were a vegetarian would have had a bad impact on her survey results.

---

[7]E.g. `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 02/11/16

Chapter 7

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

Emails with Chefkoch.de

[Chefkoch GmbH] Betreff: Liebes Chefkoch-Team, ich schreibe zur Zeit meine Masterarbeit in Informatik über das Thema, wie man aus reinem Text Zutaten extrahiert. Als Mot...

Von: "Chefkoch GmbH" <support@chefkoch.zendesk.com>
An: Stu121256 <stu121256@informatik.uni-kiel.de>
Datum: 12.01.2017 12:29:03

##- Bitte geben Sie Ihre Antwort über dieser Zeile ein. -##
Ihre Anfrage (18729) wurde aktualisiert. Um zusätzliche Kommentare hinzuzufügen, antworten Sie auf diese E-Mail.

Kundenservice (Chefkoch GmbH)
12. Jan., 12:29 CET

Sehr geehrter Herr Knauf,

vielen Dank für Ihre Nachricht.

Leider dürfen wir derartige Informationen nicht herausgeben. Vielleicht ist die Größenordnung für Sie interessant, dass bei Chefkoch aktuell 100 Mitarbeiter arbeiten.

Viele Grüße

Ihr Chefkoch.de Team

Impressum und rechtlicher Hinweis: http://www.chefkoch.de ist ein Produkt der Chefkoch GmbH, Rheinwerk 3, Joseph-Schumpeter-Allee 33, 53227 Bonn
HRB 18761, Amtsgericht Bonn
Geschäftsführer: Martin Meister, Arne Wolter
Tel: +49 228-286695-0 (kein Kundenservice), Fax: +49 228-207678-31, Kontakt

Stu121256
10. Jan., 17:06 CET

Liebes Chefkoch-Team,

ich schreibe zur Zeit meine Masterarbeit in Informatik über das Thema, wie man aus reinem Text Zutaten extrahiert. Als Motivation würde ich gerne unter Anderem schreiben, dass viele Rezept-Empfehlungssysteme auf Zutaten und entsprechenden Nutrion-Informationen beruhen, welche einen Mehrwert für Kochseiten haben. Desweiteren würde ich gerne ausführen, dass Kochseiten viel Wert haben. Leider konnte ich jedoch keinen Umsatz von Chefkoch.de finden und auch nicht den Kaufpreis, welchen G+J 2007 gezahlt hat.

Über eine Auskunft oder helfende Größenordnungen würde ich mich sehr freuen.

Viele Grüße,
Torsten Knauf

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.

Ahn, Yong-Yeol et al. (2011). "Flavor network and the principles of food pairing". In: *Scientific Reports*. URL: http://www.nature.com/articles/srep00196.

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. (2001). "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).

Freyne, Jill and Shlomo Berkovsky (2010). "Recommending Food: Reasoning on Recipes and Ingredients". In: *User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings*. Ed. by Paul De Bra, Alfred Kobsa, and David Chin. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 381–386. ISBN: 978-3-642-13470-8. DOI: 10.1007/978-3-642-13470-8_36. URL: http://dx.doi.org/10.1007/978-3-642-13470-8_36.

Geleijnse, Gijs, Thérèse Overbeek, and Nick Van Der Veeken (2010). *Extracting Vegetable Information from Recipes to Facilitate Health-Aware Choices*.

Henning, Beate (2008). "Daz ist ein guot geriht und versaltz ez niht". In: *Texte für Technik - Ausgabe Herbst 2008*, pp. 4–5.

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*.

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

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

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).

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4.* CONLL '03. Edmonton, Canada: Association for Computational Linguistics, pp. 142–147. DOI: 10.3115/1119176.1119195. URL: http://dx.doi.org/10.3115/1119176.1119195.

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.

Wiegand, Michael, Benjamin Roth, and Dietrich Klakow (2012). "Data-driven knowledge extraction for the food domain". In: *Proceedings of KONVENS 2012.* Ed. by Jeremy Jancsary. Main track: oral presentations. ÖGAI, pp. 21–29. URL: http://www.oegai.at/konvens2012/proceedings/07_wiegand12o/.