**Classify Notifications**

To classify the notifications, we used a different classifier for the Fire department, Police and Ambulance, because they used a different notation.

We used a Naïve Bayes Classifier Library, to classify the notifications.

We generated our own sample set, from which we removed the locations as they influenced other notifications in that area too much.

**Retrieve News Feed**

To be able to associate P2000 notifications with news articles, these articles need to be retrieved from the web.

We built a specialized web crawler that takes a number of preselected regional news websites as seeds and retrieves all articles within a certain date range. These articles can then be classified and indexed by other modules of the system.

**P2000 → News association analyzer**

Not every retrieved news article is relevant for a P2000 notification. To be able to link certain notifications with news articles, associations need to be created and ranked.

An algorithm based on association rule learning is used to create and analyze associations between P2000 notifications and news articles using their labels and metadata.

**Index Notifications**

This part takes care of gathering all the useful information about the P2000 notifications.

It uses the HTML structure and regular expressions to extract the information in the notification itself and it uses an API and a database of address data to retrieve more relevant information, like postal code for a given street and town and coordinates for a given postal code.

**Cluster Notifications**

This part takes care of clustering the P2000 notifications that concern the same incident. It applies our own partitional clustering algorithm using the date and time and the location of the notification as a measure.

If two notifications are within a 30-minute time frame and go to exactly the same location, they are clustered together.

**Collect P2000 notifications**

To create a collection of P2000 notifications, a P2000-monitoring website was scraped using a purpose built scraper. The location of the notifications in the websites DOM was hardcoded into the scraper and each notification was passed on to the indexer as a raw HTML string.

A polite page request frequency was adopted and the database can be kept up to date by recurring invocations of the scraper.

**Graphical User Interface**

A GUI was built to visualize the amount of emergency service calls in a user’s neighborhood.

An intuitive Google Maps API was used to display the P2000 notifications as pins, color coded to the type of emergency service. Clicking a pin will show related news items or tweets if an association was detected.

**Language & Spelling Detection**

Often a lot of errors are made in news items and different news sites use different style for their writing.   
We try to achieve higher level of similarity between all these articles by applying a language and spelling analyzer on this.   
This also helps for the Tweets.

**Classify News Articles**

To associate articles to P2000 notifications we need to label these articles with their subject and whether they are relevant or not. A Naïve Bayes classifier is used for this.   
A challenging task due to the vast diversity in writing styles and subjects reported on by local news site.

It is used to find Ambulance, Police and Fire Department related news articles.