

Lessons from mapping Sahel

We needed an example problem set for our current version of [Hunchworks](#) (note that this is a very early, i.e. pre-alpha version of the code and a lot of the cool Hunchworks features aren't in it yet). The UN's main use for Hunchworks is to gather up the weak signals that people put out about emerging development crises - those small hints that something isn't right that appear all over the world before they coalesce into 'obvious'.

Awareness of development crises can happen very quickly. One minute there are whispers of a potential problem – a chat here, an email or text asking for a bit of data there. And then a tipping point appears and there's suddenly data everywhere. And we have a great example of this happening just at the time that we're demonstrating Hunchworks to the UN General Assembly.

We had one of these serendipitous test sets before: we tracked the Horn of Africa crisis emerging across newsgroups as one of our early will-this-work paper exercises (this btw is also why we're suddenly interested in data mining googlegroups). But the Horn of Africa crisis is well established in the public eye now, and there is both too much online data on it to pick out the early weak signals, and many of the early traces (e.g. anecdotes and messages) have been lost in both human and machine memories (yes folks, not everything on the Internet is logged). But there's a new thing starting to happen (which is potentially very very bad for the world and certainly for the people caught up in it) - over the last month or so there have been mysterious messages here and there about something starting to happen in Chad and Niger, across the African region known as [Sahel](#).

So this week we started [collecting information about Sahel](#) and [turning it into hunches and evidence](#) in Hunchworks. First we had an email from a trusted colleague containing potential places to look. Then an Internet search for news and background information, followed by more digging across the Internet (including recent reports from the UN and other NGOs), a colleague searching in the food-related UN agencies and a Twitter search for hashtags and interested parties.

We could have got a lot more information faster and with more insightful comments into it if we'd [crowdsourced its collection](#), but [the EOSG](#) couldn't be seen getting involved prematurely (i.e. before the dedicated HLWG team) in this crisis. We could have also involved more people in a non-public search if we'd had Hunchworks at its Beta testing stage, but doing the first search by hand is a sane early stage test that exposes bugs early to a small number of people before a larger group (e.g. the Alpha and Beta testers) get annoyed by them.

So what did we learn about ourselves, our information and Hunchworks from this exercise?

We ran the exercise using [a spreadsheet](#) (no, we can't just do this for hunches because it will quickly become overwhelmed - see below). Its first worksheet was a list of hunches: this was

quickly populated with a mix of hunches and evidence for hunches that took some time to separate out. We also discovered that the evidence that we gathered often contained new places to look for evidence, suggested new problems that should be proposed as hunches and spawned a whole pile of other evidence-gathering activities.

- Lesson 1: people confuse hunches and evidence. Sometimes evidence is posted on the hunches list; other times hunches are posited as evidence to another hunch.
- Lesson 2: evidence generates hunches. For example, we realized that a hunch about a famine in Sahel also contained hunches about famines in Mali and Chad that the country teams there needed to investigate.
- Lesson 3: evidence generates evidence-gathering activities. We ended up with a to-do list linked to each hunch.

Some things were confirmed for us. We suspected that we could map the connections between hunches as though the hunches were propositions in a reasoning system. We also suspected that there were a set of basic search actions that we would do at the start of most hunches, that some of them would be ongoing (i.e. to catch new information being added to the Internet) and that we could automate many of these. Yes.

- Lesson 4: when we draw graphs using our hunches as nodes, the links between nodes look suspiciously like the links in semantic networks. This should come as no surprise to anyone working on linked data.
- Lesson 5: Google searches, news searches, twitter searches, UN report searches and emailing around likely suspects are obvious first things to do on any new hunch.
- Lesson 6: We can automate some of the above searches, especially if we have search terms (e.g. Sahel) and tags to start from. Our hitlist for Sahel was: twitter stream, food price, migration from/to Sahel and news monitoring agents.

We've tried to build a system that doesn't need much management or moderation. We might need to revise that: in the initial excitement of chasing up leads and links from the original hunch, it was difficult to maintain momentum (e.g. amount of evidence added) and completeness at the same time. I had to do a lot of reading and editing – both to disambiguate hunches and evidence as discussed above, but also in generating tags, thinking about the links between hunches and managing the list of actions that happened most times we added any evidence. Some more lessons from this:

- Lesson 7: Hunches have information-gathering actions attached to them.
- Lesson 8: Once we get textual evidence, it's pretty easy to create tags from it.

And then we've got some very specific lessons about the system.

- Lesson 9: if two hunches are related, they probably need the same people involved in them.

Can we start the “involve x” list of one from the other?

- Lesson 10: Some places had 2 locations, e.g. migration from Libya into Chad.
- Lesson 11: We have the same problem as crismappers with location accuracy, e.g. sometimes we want to mark a region rather than a single point on the map.
- Lesson 12: Using tags brings a set of questions about how we find related things again. This is the same issue we've seen in crismapping and Twitter feeds, and we have tools that can help with this.

There are more lessons learnt, but we're somewhat busy today. More soon.