Forprosjektrapport

Hovedprosjekt Gruppe 15

Erlend Gunnesen, Lars Sætaberget, Are Inglingstad, Marius Maudal

25.02.2014

Innholdsfortegnelse

1. Introduksjon

Vi er en gruppe på fire på Høgskolen i Oslo som har fått en oppgave av Retriever om å lage et bildeanalyseverktøy.

1.1 Medlemmer:

Lars Sætaberget
Erlend Gunnesen
Marius Maudal
Are Inglingstad

1.2 Oppdragsgiver:

Retriever Norge AS
Langkaia 1
0101 Oslo
+47 229 10 350
post@retriever.no

1.3 Kontaktsperson hos Retriever:

Silje Charlotte Brekke
Morten Johnsen

1.4 Veileder:

André Lincoln Read
+47 412 68 262
andre@asio.no
www.asio.no

1.5 Presentasjon av arbeidsgiver

Retriever AS er ett firma som jobber med innhenting av nyhetsinformasjon for bedrifter. De henter inn informasjon fra websider, aviser, tv-sendinger og lignende. Hensikten er at bedrifter skal kunne holde seg oppdatert på all nyhetsinformasjon. Firmaer har egne søkestrenger tilpasset sitt informasjonsbehov.

Retriever ønsker å bli det naturlige valget for bedrifter når det kommer til informasjonsinnhenting i Norden.

2. Sammendrag

Formålet med prosjektet er å automatisere uthenting av tekst fra tv-bilder.

Løsningen vil bli skrevet i java. For å få til løsningen må vi ha flere deler som jobber sammen. Data blir hentet fra bilder. Deretter behandles bilde for å fremheve kontrasten til teksten. Dataen sendes så til Tess4J som er et dekodingsverktøy for java til Tesseract. Informasjonen blir så sendt tilbake til en leksikalsk analyse, som skal kontrollere at teksten vi henter ut er korrekt. Ved feil på ord i teksten vil dette bli markert i XML filen som lages.

3. Dagens situasjon

Retriever leier i dag inn tjenester fra ett eksternt selskap for å få skrevet inn artikler manuelt i XML format. I dagens automatiserte samfunn er dette en veldig primitiv løsning. Det medfører også ekstra kostnader som Retriever ønsker å bli kvitt.

Ved vår løsning kan da Retriever spare betraktelig med både tid og penger.

4. Mål og rammebetingelser

4.1 Mål

Målet med dette prosjektet er å automatisere tekstgjenkjenning på bilder. Oppdragsgiver ønsker en løsning som fungerer med 100% sikkerhet. Løsningen skal kjøres på kommandolinje i Linux uten GUI. Gruppen kan selv velge programmeringsspråk så lenge det fungerer i Linux. Programmet skal kjøres så raskt som mulig.

Hvis gruppen ikke klarer å få til løsningen 100% skal det være en feilsjekk som informerer om hvilke deler av XML file som er feil. Feilrapportering må være veldig sikker da feil i dette kan medføre at feilrapporteringer ikke blir håndtert.

Det viktigste for Retriever er at programmet skal fungere med deres systemer og ha muligheter for videreutvikling av enten gruppen eller oppdragsgiver.

4.2 Rammebetingelser

4.2.1 Prosjekt

Tid og rammebetingelser:

- Statusrapport ble ferdigstilt 25.10.13
- Prosjektskisse ble ferdigstilt 21.01.14
- Forprosjektrapport ble ferdig 24.03.14
- Prosjektet skal være ferdig 27.05.15
- Presentasjon av prosjektet foregår tidlig i juni

Egne mål og rammebetingelser:

- Ved dette prosjektet skal vi lære seg åssen systemutvikling fungerer i arbeidslivet. Hensikten er at vi skal få praktisk erfaring.
- Vi må ved dette prosjektet sette oss inn i veldig mye nye systemer og tekniker som vi ikke har hatt erfaring med fra skolen.

4.2.2 Teknisk

Programvare:

- Programmet skal kjøre i Linux-miljø
- Programnavn skal utvikles i Java
- Programnavn skal utvikles i standard JRE 1.7

Annet:

- Oppdragsgiver vil supplementere med testmateriell som skal brukes i et privat testmiljø. Reel testdata er viktig for å få ett best mulig testmiljø. Vi kan da få testet hvor bra programmet håndterer den store mengden med data.
- Oppdragsgiver vil gi oss en arbeidsplass på deres lokaler for å kunne ha god oppfølging av kontaktpersonene.

5. Løsninger/alternativer

Én løsning vi ser for oss er en wrapper bestående av en xml-skriver, filfeeder, tekstsammenligner, bilde-behandler og -analysemotor.

5.1 Tesseract

Vi ser for oss en løsning rundt opensource OCR-motoren Tesseract. Da vi planla mulige framgangsmåter/løsninger så fant vi ut at Tesseract ville gi oss de beste tekstgjenkjenningsmulighetene, samtidig som den er opensource og har vært i bruk lenge. Det vi ønsker å gjøre er å bygge vårt system rundt denne motoren (en wrapper), som laster inn bilder, behandler de med skalering/binærisering/kontrastjustering og mater de til Tesseract.

5.2 Leksikalsk Analyse

Videre så må resultatene vi får behandles videre i en leksikalsk analyse og skrives til XML. Grunnen til at vi må ha en leksikalsk analyse er fordi flere bilder fra en nyhetssending kan inneholde den samme informasjonen flere ganger. Planen blir å kjøre ordboksjekk på ordene vi får fra tekstgjenkjenningsdelen.

5.3 Ytelse

Det stilles også krav til ytelse, men vi tror at selve tekstgjenkjenningen vil ta lengst tid, og derfor kan vi skrive vår løsning i et språk som f. eks. Java, og evt. oversette til et raskere språk (C++) senere hvis det er tid og behov. Et annet viktig poeng er at Java er det språket vi er mest komfortable med, og det er viktig at vi får ting til å fungere tidlig nok, slik at vi har tid til å debugge og rapportere. Vi har naturligvis tenkt å lage støtte for multithreading, siden det vil redusere bildebehandlingstiden betraktelig. Skopet i oppgaven er lite, men vi må bruke mye tid til å optimalisere treffsikkerheten ved å justere bildebehandlingsalgoritmene.

6. Analyse av virkninger

Vi har bevisst vært litt vage når det kommer til løsningen, da vi beveger oss i meget ukjent farvann. Vi er avhengig av å finne de spesifikke løsningene til problemene mens vi jobber.

6.1 Tesseract

Hovedgrunnen til at vi skal integrere Tesseract i vår løsning er at det er meget tidsbesparende for oss å ikke lage en egen løsning. Tesseract er som tidligere nevnt open source og kan brukes til kommersielt bruk som er en stor fordel hvis oppdragsgiver velger å bruke vår løsning.

6.2 Leksikalsk Analyse

Den leksikalske analyse vil bli laget med utgangspunktet i norsk, svensk og dansk. Det er viktig at vi koder den så den lett har mulighet for utvidelse til flere språk. Analysen vil antagelig bli en av de delene med mest utvidelsesmulighter. Vi vil mest sannsynlig ikke ha tid til få på plass større ting som setningsanalyse i koden.

Målet til analysen blir å avdekke dobbelt informasjon og feil i ordene. Vi kan risikere å miste informasjon hvis den leksikalske analysen ikke fungerer korrekt.

6.3 Ytelse

Ytelsen er et punkt som alltid kommer til å ha forbedringspotensial. Vi har i vår løsning dekket fremtidige endringer i programmet som kan øke ytelsen.

Disse punktene konkluderer tankene vi har hatt rundt prosjektet. Sammen med kravspesifikasjonen utgjør dette dokumentet en grov skisse om hvordan vi tenker at programmet skal bli laget. Vi er avhengig av god dokumentasjon av løsningene vi finner underveis.